Developer Keynote: Build the Future with the Platform

Auto-transcribed by https://aliceapp.ai on Friday, 20 Sep 2024.
Synced media and text playback available on this page:
https://aliceapp.ai/recordings/w4cH8bvYBmpgPn6RKCIrsuMiL2X_J7oE.

Words	8,707
Duration	00:51:09
Recorded on	Unknown date
Uploaded on	2024-09-20 00:54:52 UTC
At	Unknown location
Using	Uploaded to aliceapp.ai

Speakers:

Speaker A - 13.13%

Speaker B - 30.29%

Speaker C - 25.76%

Speaker D - 30.83%

Notes:

- The history of software development is really the ongoing pursuit of higher levels of abstraction. The Salesforce platform abstracted these basic services and kept adding new ones three times a year as technology continued to evolve. The next level of abstraction is to bring all the data from all these data sources together and expose it as if it was a single data source. That's what data cloud does.

- There's been a massive shift over the last twelve months in the way that a lot of you are actually building with data cloud. The other major shift that we've seen in the last 12 to 18 months is obviously generative AI. At Salesforce we're really focused on building tools and frameworks for you as developers.

- Model builder is a tool that we introduced a couple of years ago now to help you bring predictive and sort of static models into your Salesforce orgs. We recently announced the ability for you to create your own predictive models based off of the data that sits in data cloud. Use this to identify which models may be good for you as a builder.

- In Apex we have a new models API method as part of our APIs in Apex. Each prompt can be set to use a specific model that we provide. You can execute this from anywhere, your mobile apps, your websites. This is perfect for developers wanting to build their own generative AI apps.

- Ananya is the product manager for agent force for developers. She says this is the most exciting time to be a developer within our ecosystem. All of these announcements are things that you've been looking for, she says.

- All right, thank you so much, Anania, Stefan J and Alba. We saw how to build so much with data, AI and tools. To learn more, we created that developer quest. Take your phones out and scan that QR code.

Speaker A 00:00:00

All right. All right. Hello developers, and welcome to the developer keynote. You know, we always say thank you, you know that. But today I really want to thank you for being here. I know it's the last day of Dreamforce, it's the last keynote, it's the last session. But if you know the speakers that we have in store for you, you know that we kept the best for last. I also want to thank you for all the innovation that you all drive on this salesforce platform. This is an incredible platform for developers and that's really what we want to talk about today. Now to really understand the power of that platform, I want to go back in time just a little bit because if you think about it, the history of software development is really the ongoing pursuit of higher levels of abstraction. And this is actually the architecture diagram of the first, uh, application that I ever built. Probably in the fifties at this point. And at that point there was no abstraction at all. You had to build everything in a single codebase, low level. And maybe some of you are with me on this and experienced that low level UI logic, uh, low level business logic, and most of the code was really low level data access. This is why the first major abstraction that we built was data. With the kind of widespread adoption of database servers that did really the heavy lifting to access your data so that you could focus on what really matters, um, about your application. Now in the late nineties, someone that you may know had that crazy idea of hosting enterprise software in the cloud. And that was essentially the second major level, uh, of abstraction. It was infrastructure. We moved servers in the cloud that made it easy to scale and that made it easy to upgrade software and much more. Now I know there are many vendors today that provide cloud computing platforms, but even with these platforms, I felt myself reinventing the wheel all the time, writing that same old code. And maybe you have been there mostly around data entry. That's the code that we were writing. Create, update, delete. That was it. And that was my first introduction to the Salesforce platform. Because unlike other platforms, the Salesforce platform abstracted these basic services and kept adding new ones three times a year as technology continued to evolve. And with that I was able to really focus on coding, differentiating features and innovation. Now, what was the next level of abstraction? Just want to ask a quick question. I assume that most of you wouldn't think about building your own database server before you start building your own application, right? Usually you don't do that. Well, I will argue that for the same reason no one, no one should build their own CRM applications before they start building their application because these problems have been solved. They have been commoditized and I will argue they've been solved pretty well by Salesforce. So that's the next level of abstraction, these CRM applications that you can build on top of. Now, meanwhile, these developers are not a platforms. They are still reinventing that wheel. And while they are doing that, they are not delivering, they are not coding these differentiating features and innovation. But you know what you are, you are on um, the salesforce platform. Ok? Now that we work with multiple applications, we are no longer working with a single simple data source. We are actually working with dozens of data sources. So the next level of abstraction is to bring all the data from all these data sources together and expose it as if it was a single data source. And that is what data cloud does. And that's what data cloud does really, really well. All right, so now we are almost there. Is there anything else that we should abstract? How about AI and models? This is also something that you don't want to diy and that's why we built it inside the platform, so that your application doesn't depend on a single vendor or a single model. If tomorrow there is a better model that's newer and better, we will just add it to the platform so that you can use it. But you know what? These models, they are more than just about generating content because today they can reason and they can orchestrate tasks. And that gives the salesforce platform another unique advantage around something that we've been talking a lot about this week. What have we been talking about this week? Agents, right? Agents. Have you heard about agents? Agents, right. Okay, so imagine breaking down these CRM applications into small pieces, each of them encapsulating some discrete logic, right? It could be create a lead in sales, changing the delivery address in service, make a payment in commerce. And then the way these applications of the future, the way these agents work is that when you give them a task, they will reason over all these building blocks and the reasoning engine will then identify the best building blocks to use to deliver the solution. This is Agent Force. This is the combination of out of the box agents and then the tools that you need to create and customize agents. And all together, this is the Salesforce platform. This is what we want to talk to you about today. And to do that we will use a sample app that we call Coral cloud. And today I'm super excited to share that. We made that application available for you on Gita as of today you can install it from GitHub. But get this, as you do that you can get your own provisioned with data cloud and Agentforce. This is something that you have been asking for a long time. Can we get our hands of it? And the answer is yes, you can. So scan that QR code and get the application. All right, so now it's time to start and to see how we can build a future with the Salesforce platform, um, using data AI and tools. And we will start with data. Please welcome Jhurst SVP product management.

Speaker B 00:07:50

Thanks, Christoph. Hello Salesforce Developers. Get some energy. It's okay. My name is Jay Hurst. I'm a product manager here at Salesforce and I've had the honor of being employed here for a little bit over 20 years. And I have to say that I am more excited now for our developer ecosystem than I have ever been before. That's because we have done so much work with data cloud. Now for about the last two and a half years, my teams have been focused really to bring the power of data cloud into the core platform. And hopefully you've started to see some of the results of that through things like uh, CRM enrichment with data cloud related lists and copy fields and data cloud triggered flows and data cloud reports. And that's all been possible because of that democratized data layer that Christoph just mentioned. We've built a unified metadata layer so that wherever your data lives, Salesforce can reason about it. And this has allowed us to really accelerate what we're doing with data cloud. Things like data cloud one, uh, our uh, sub second real time platform or generative AI features like vector database and our new hybrid search. All of these are possible because of the work that we've put in. Now, seeing it on a slide is great, but I know as developers you want to see an actual demo of this stuff in action. So I'd like to welcome Alba Rivas who's going to be the demo driver today as we go through Coral Cloud. So this will be a whirlwind tour of data cloud. But like Christoph said, this is all available in GitHub. So don't worry if you missed something, you're going to be able to get your hands on it after the show today. But we're going to start here in uh, Coral Cloud's data cloud home. Now, Coral Cloud is a uh, boutique resort in Hawaii, very customer focused and they really are excited to bring the power of data cloud to their agents and customers. But inside of the data cloud home, this is where you will provision and manage data cloud. It's kind of a one to one mapping with a single CRM.org. and the first step is to tell data cloud where your data is at. And we'll do that through data streams. So as we see on the screen, we have a number of data streams available. We um, see a number of contacts from salesforce, CRM orgs. Uh, we see a guest, uh, table that's being brought in from AWS. Could uh, be other cloud vendors could ingest or zero federated copy. And we have tons of other connectors, hundreds available. Or if you are a mulesoft customer, any Mulesoft connector is available as well. Now, once we've created a data stream and told data cloud where your data is, a corresponding data lake object is created. Now this is a one to one mapping. And the data lake object, or Dlo, is data cloud's representation of that data. So just like we saw the guest file, uh, over in AWS as a data stream, we see the same guest as a profile category in data cloud as a DLO. If we open that up, what we will see is the mapping or the fields that were brought over the schema, all of that ingested from AWS. And we also have the ability to map. Now, mapping is the next stage of data cloud. You take your data lake object and you map it into a data model object. And this is where we start to really see the power of data cloud. This is the harmonization step. So we have this guest table, we're mapping it to a couple of different DMO's, including the individual object that you'll see on the bottom. Right. And we would do the same with all of those contact tables that we brought in or any other person type of table. So we've gone ahead, we've told data cloud where your data is, we've modeled it, we've harmonized it, and the next step that we can do is called identity, uh, resolution or unification. This is how you tell data cloud across all of these different data stores. Here's how you match the records together to identify they're the same person or account. Now, we have a rule here for fuzzy name and normalized email. What that means it's actually going to look at the first name kind of fuzzy. Match it on all of those tables, an exact match on the last name, and an exact match on an email address. If all of those are true, data cloud will know that it's actually the same person. So if we go ahead and, uh, take the next step after unification. Let's see this in action. First we're going to run a query on that individual DMO. So we'll run it looking for any record with the last name of Rodriguez. And we see four individual records, one from each of the CRM orgs that we mapped and one from that, uh, AWs s three bucket. Awesome. Now, if they are actually Sophia Rodriguez and they match correctly, the unified individual should match to a single record. So we'll go ahead and run that same query against the unified individual. And indeed we do have a single Sofia. So across four different data sets, three of them being different salesforce, orgs, one of them being Amazon zero copied, we now know that it's the same person. And since we know it's the same person, all of the connected data to those various records can be related to this one unified individual. And this is where we can really start to see the power of data cloud. So we have our queries. Let's see how this looks when we want to share the data. So we've now entered pixel harbor. This is an affiliate hotel, uh, of coral clouds, and it is in the data cloud one application. Now, data cloud one is a new feature we just introduced which allows you to do bi directional multi.org sharing of data cloud data. Previously you could ingest data from multiple Crmsheen. Now you can expose and share with multiple CRMs. So to prove that, if we jump into the query editor, we'll go ahead and run that same query on the unified individual and we should see one Sofia Rodriguez record. So this is now even more powerful. That same unification we saw in the previous, now available in a completely separate, all declarative, no code written so that you can actually start to spend your time in code where you need. And speaking of code, that's going to be the next step. Just like any other good data platform, data cloud has open APIs that will allow you to run SQL queries. But we want our Salesforce developers to use Salesforce code where, uh, they need. So inside of Apex, because we have that unified metadata layer, all of the dmos are now exposed straight to cycle. So we can see a number of queries here on the unified record id based on a local contact. We can take that and we can get all of the related individual records and then we can even join that with a table of reservations all exposed in the standard sock you would expect like any other custom object. So we've now actually federated a query across multiple external tables and brought that data together so you can actually see a unified view of a guest's entire reservation history across all of your data stores. This is huge. Now, running these queries is great. As developers, that's how you're going to explore. However, if you want to put something into production, you might want a little more performance. And so rather than running three individual queries, I think we can do a little bit better. And that's where data graphs come in. Data graphs are pre calculated, materialized view of your tables, so you have the ability to create them and structure them for the data you need. And we're proud to say, as of now, data graphs are part of our sub second real time engine. So anytime data is added to that graph or updated, it will, uh, be available to all of your activation points in under a second. Extremely huge. So as we go to the next step in the data cloud, or, uh, data graph builder, what we would see, just like we built those three separate queries, we're modeling out the same structure. We can do this because of that unified metadata layer. Salesforce understands relationships between salesforce tables. We also have the opportunity to select or deselect any fields we may or may not need so that we can quickly retrieve all of the information and provide that great guest experience. So we'll jump right into a, uh, view of what a data graph looks like. And so if we look at this, you'll see an entire JSON payload that will give you the unified individual, the individual and all of the associated reservations in a single payload that can be used wherever you need. And all of this data is available via APIs. What Alba is showing us here is our postman collection for the connect APIs on data cloud. And you can see a simple data graph endpoint that you can use wherever you need. When I say wherever you need, that's in any API, whether it's LWC, Apex, or anywhere else. Now, we've been able to model our data, map our data, ingest it, harmonize it, expose it where we need. We're ready to go. The next step. Like any customer focused.org, coral Cloud has a ton of unstructured data around them. A lot of that data might be some of those travel guides that you see at the front desk of what adventures should you take when you're visiting their hotel. Some of the adventures might be a little more, uh, aggressive, I'll say, than others. And, uh, in Hawaii, we want to know about the volcanoes and what to expect. Handing these pamphlets out and telling your guests to search through them, that's not what we want to do we want to expose this through agent force? So this is where, uh, semantic search comes in. Semantic search is really good at looking across the broad sets of data and finding intent rather than just the specific keywords. We see that we have a couple of, uh, searches or indexes already built, but we'll go ahead and create a new index. Once we do that, we'll see a couple of options. We have a vector search and a hybrid search. Hybrid search now in beta. But what is that, you might ask? Well, like I said, vector search is really good for searching across semantic data, but it's not that good for searching keywords. So product codes or potential hawaiian volcano names might not show up as. Exactly. That's where hybrid search comes in. With hybrid search, you are able to create both the vector index as well as a keyword index and use those together to provide better results. So we'll go ahead and start and building our index on the travel guides. DMO travel guides is where we uploaded all of those pamphlet, uh, PDF's in AWS. First step is our chunking strategy. The chunking strategy is how you tell data cloud. How do I split this file up into reasonable pieces? Different file types might want different chunking strategies depending on how that data is laid out. Next, we'd go into our vectorization to choose the embedding model we want to use next. It would be, uh, adding any additional fields you'd want to filter for your search. And then finally we would review and build that, uh, index. That could take a little time, so we're going to go back to one of those two that we used before our travel guide hybridization. Now to put it into action, just jump straight into our query editor again and you'll see we're going to run, ah, against the travel guide index. This isn't the hybrid one, this is just the regular index. Adventure activities in Hawaii as Alba runs that we'll see. Great, a number of scores coming back. Now, what's a score? A score is how close, uh, of a match is it? It's pretty complicated, I'll be honest. I don't understand it completely. But what I've been told is the closer to the number one, the more accurate the search. So we have some pretty accurate search results here. But I would like to see more information. So let's go ahead and join that score with the chunk table that we just created. We run the same query. Now we see the scores along with information of what Chunk did it match? What file was that chunk from? What type of file size, uh, and, uh, uh, content type is it, etcetera. So now we're starting to get some confidence. But you can see we have a couple of really, really close matches at 0.83. I think we can do better with that hybrid search. So we go ahead and run the same query just against the hybrid chunk instead and hopefully we'll find an exact match with the number one and we do. Boom. So we have a lot of confidence in our search results here. Now, since this is part of the platform, it's part of Agent Forest. We can use this anywhere on the platform, including through our prompt builder with the new search retriever. So now you are able to bring these search results right into your prompt templates and leverage those across anywhere you're using prompts. All right, we're almost at the end. Let's put this into practice and actually enter the new agent builder. Inside of agent builder, Alba is going to run our simple, uh, request. Show me some top sites on the big island that's going to send it over to our, uh, hybrid search. It's going to start running that check again like we saw. It's going to do any of the questions and setup that we did and it hopefully will bring back our full result set. And here it is. We see that the top five sites for us to visit in Hawaii, all pulled from those pamphlets, uh, that we had at the front desk. And that top one interests us because we're about to go volcano hiking. I guess we'll go ahead and ask what do we need to do to prepare to go hiking? So we'll run the same query, what precautions should we take again going to that same search model, uh, that we built inside of agent builder, uh, running through the same topics and ultimately should tell us what we need to do to prepare. And there it is, as we see, stay on mark trails, dress appropriately with sturdy shoes, volcano proof I hope. And with that, we will complete the data cloud journey. Hopefully you guys see why I'm so excited, the power of data cloud inside of Salesforce, why this is such an exciting time to be a developer at Salesforce. And hopefully you can also see that data cloud is not just a simple data store. Data cloud really is the backbone for everything we are going to be building on this platform for the next decade. And with that, I'd like to thank you for your time and introduce Stefan.

Speaker C **00:23:09**

All right, thank you very much, Jay. Uh, one of my favorite parts about these conferences, whether it be Dreamforce, whether it be some of our amazing community conferences, it's talking to you developers about the technology and the direction that Salesforce is going. And there's been a massive shift over the last twelve months in the way that a lot of you are actually building with data cloud. And just hearing about the things that you've been doing with it really inspire me as a developer. Now, the other major shift that we've seen in the last twelve to 18 months is obviously generative AI. It's had a massive impact on just about everything I build, whether it be what I'm building, the apps that I'm building, the tools I'm building with. I mean I feel like everything's pretty much changed and that brings a whole load of new challenges thanks to security and privacy and all the things that we really need to keep in mind when we're using these tools. And at Salesforce we're really focused on building tools and frameworks for you as developers that allow you to bring generative AI into everything that you're building in a safe way. Things like the trust layer, agent force, model builder, the models APIs, these are all things that I'm sure you've seen and maybe even used this week at Dreamforce. And so what we're going to do is go to a demo and really abstract these tools into smaller bite sized chunks around how they impact you as developers. So again, thank you Alba, let's take that to the demo. And so this all really starts in model builder. Model builder is a tool that we introduced a couple of years ago now to help you bring predictive and sort of static models into your Salesforce orgs that are creating predictable results around classification, maybe some type of indexing or sentiment analysis. Okay? And we recently announced the ability for you to create your own predictive models based off of the data that sits in data cloud now. But I want to focus today on those generative models, those foundational models in our now it uh, seems like just about every other day a new vendor is creating a new model and launching it into the marketplace. And whenever we find a model that's fit for purpose, for the context of CRM, we want to make sure that it's available by default out of the box. Now you can see we have models available from OpenAI, we have some of the Azure implementations of OpenAI available. And recently we've launched our very first Salesforce hosted model available in our tools and that is through Amazon bedrock. And that's going to be Anthropic's cloud three haiku. Okay? That model is actually being hosted on hyperforce inside of the Salesforce Trust boundary. So when you're sending calls out to the large language model, you're no longer sending data or sending your prompts in flight to a third party. It's happening in the context of the platform itself. Now a lot of organizations really rushed to to building and training their own models, fine tuning their own models, and we want to make sure that those are accessible to you as well inside of our tools. And so we've added the ability for you to create your own connection to a model that your company is hosting themselves, whether it be from Azure, OpenAI, your own fine tuned OpenAI model, something you're hosting in Google, Vertex or on Amazon, bedrock yourself. All you have to do is connect that into model builder. Now we've already connected a custom anthropic sonnet model that we're hosting on bedrock ourselves as coral clouds. Now when we connect these models, okay, we can set the settings so that we can identify how many responses we want to test. We can set a custom temperature mapping for this and then we can test it with our prompt. Now in this case you would probably be fine tuning this model on more business specific data for your use case. And so we're going to test it out and ask it to write an agenda for our beach bootcamp because that's obviously what Corel clouds would train it on. It's sending that prompt across through your credentials to your endpoint to get a response back from the large language model. So it's hitting your own custom model that you've attached into Salesforce and it's going to come back with that detailed agenda that it's generated. That means that you don't have to worry as much about things like data masking because this is being sent to your model that you may be training on based off of the responses because it's a custom endpoint that you've brought to our IO tools yourself. Okay, so we've connected a model, we've looked at what's already in existence. Now, I am no AI researcher or expert and so I couldn't tell you which model is fit for purpose for your company, but we've got a great AI research team who has ran an LLM benchmark around a lot of publicly available open source and commercial models to identify which models perform better in the context of CRM and our Salesforce apps around accuracy, cost, whether or not they actually follow the instructions that you're sending their completeness, etcetera. And so you can use this as a tool to help you identify which models may be good for you as a builder. Okay, we have models connected. How do we use them? Well, I want to go over to setup in Salesforce and just make sure that we have a feature turned on. Okay. One of the core features to sort of the trust and safety that we provide around our generative AI tools is that we're logging all of our feedback and audit data in data cloud if you have this feature enabled. And so every time someone gives feedback on a response, every time a prompt is sent to a model, we're storing the prompt, the response, the resolution, everything that comes through inside of the feedback store. And we can use this in prompt builder. You've probably seen this a million times. You've probably used it, you've probably written this prompt. So I want to focus on the actual models themselves. Each prompt can be set to use a specific model that we provide, or the custom model, in this case, that sonnet model that we connected through bedrock. And what's important to note here is that every prompt may respond differently to different models. And what we want to ensure is that whenever this prompt is executed, it uses the model that's been designated in promptBuilder. Okay, once you designate the model here, that is what it's going to run in. So if you're running an app in maybe agent force that's using a specified API or a specified model to run, it will reach out to the prompt and use the model designated for the prompt to get that insured response. Okay, now that we have connected this, we can talk about actually using them. We all know that we can execute prompts on fields to create emails in Apex and flow, but prompt template invocation is available via the Connect API in our suite of rest APIs. So you can take that same prompt we were just looking at to generate those experience descriptions. And you can execute this from anywhere, your mobile apps, your websites, and you can configure things like the temperature and which prompt you want to execute and pass in any parameters. And when we run this, it's going to take that prompt. You're accessing it off platform, but still sending that prompt through the trust layer, doing the secure data retrieval, and getting back that safety score as well. So we can see the toxicity score, we can see anything that we may want to flag before we show it to users and then the generated response. That's sort of our platform capability around prompts and grounding. But as a developer, I want to actually use these APIs myself. I may not want to rely on promptbuilder or the platform capabilities to do the grounding and do it myself. And that's where the models APIs come in. Now, this is a lightning web component that we created that uses the models API to do something a little bit more complex than we would do with a prompt template. And so we can configure this ticket generator and we can send a prompt to the models APIs directly, and we're going to generate a series of tickets for this event. And what you can see here. Great, we've got a response. What we've just done is called our custom model directly via the models API to generate this SVG. We can export it, we can print it, and let me show you how it works over in a lightning web component, we're dynamically constructing the prompt just in our JavaScript code. But what's different here is we have more control. We can use dynamic and conditional rendering and adding more content and context to the prompt. We can call out to a third party JavaScript library to generate the QR code and supply the QR code to the ticket. We're not relying on a large language model to generate a QR code. We're giving it everything that it needs to put all of that content back together without some of the safeguards that come with a capability like promptbuilder and the prompt templates themselves. Okay, how do we access this? Number one, in Apex we have a new models API method as part of our APIs in Apex, where we can call that generations API. We can pass in the body and we can generate a response. In this case, we're using that custom model and specifying it directly in Apex. Now we're also storing the invocation in the generations IDK because we want to be able to log feedback against this as well. And so if you look back at that lightning web component, we have that thumbs up and that thumbs down. We can click on that. And in our custom UI we can use the same feedback store that we're using in the apps we're building on platform. So now going back over to Apex, you'll be able to see here on that same models API module. We also have the ability for you to submit feedback based on these requests and throw them into the data store. This is perfect for isvs who are wanting to build their own generative AI powered apps and not worry about some of the restrictions of things like promptBuilder. You can use this API directly in your packages and the feedback mechanisms in your apps. Now, where does this data go? Okay, I'm going to show you quickly in reports because all of this data, cloud data is available in reports. First I want to show you the user feedback. So every time there's a thumbs up, a thumbs down, comments on your request, okay, all of that data is available to you on the platform. And then, same thing with the generations. What we'll be able to see is a list of the applications that are triggering these generations. We're going to be able to see whether they've come from field templates, from sales emails, anything on platform. You can see the prompt that was sent through to the large language model. You can see the masked prompt. In some cases you can see the response text. And you get all of this data available to you on platform M in the DMO's. See, you can see here's our ticket generation that we've been sending through to try and generate those svgs. Okay, now this data isn't just available in reports, obviously, as Jay showed us that DMO's are available in sockle, we can open up Socko builder and we can get access to this directly. So if you're building any complex UI around this feedback data, if you want to create an interface for your users to manage this, if you're building custom apps, you can query this data directly and start to build on top of it. And one last thing. The models APIs are all available via our rest API as well. So we now have all of the models APIs documented. So if you're wanting to build custom apps that use the trust layer, use all of our data masking and toxicity detection in your apps, you have access to not only the generations API, but chat completion. If you want to create a custom chat interface, you can generate embeddings and submit feedback. And that's just a glimpse at how we're building this from the ground up. The tools that we're using to build promptbuilder to build agents are all available via our tools and APIs that we're exposing to you as developers. I just really can't wait to see the way that a lot of you start to build and bring this stuff into your apps. Thank you very much. And now we're going to bring up Ananya to talk to us about those tools.

Speaker D **00:35:04**

Awesome. Thanks, Stefan. Hi everyone. My name is Ananya and I'm the product manager for agent force for developers. Now, you all just heard from Stefan how we're really bringing AI into our apps and making that possible for you. I am super excited to spend the next few minutes talking to you all about how we're making it easier than ever for you all to build on the platform. So I've been working in the developer tools and experiences area for over five years now, almost six actually. And I can say that, hands down, this is the most exciting time to be a developer within our ecosystem. And why is that? Because we've been listening to your feedback, and all of these announcements that we're here to tell you all about today are things that you've been looking for. So, first up, we have local development for LWC. This was the number two ask from our last LwC State of the union, and we all want to be a little bit faster and have those faster feedback loops when we're building our lwcs. So we're excited to announce that local dev for LWC is officially in beta, and it's available for you all to try out today. Now, second, we have my personal favorite, Agent force for developers. Agent Force for developers helps you be more productive with the power of AI. So not only are you integrating AI into your apps, but you can leverage Genai to help you be more productive as well. And the big announcement that we have for that is that agent force for developers is officially in GA today, and it's free, so you can start using it right now. Now, last but not least, we're also making sure that as you build your code and you build your applications, they're performant and scalable. And so with that, we have both Apex guru and scale center in GA today as well. You can give a round of applause for that one, too. All right, so again, we know we're all Salesforce developers, so let's see all these in action thanks to Alpha. So the first one that I want to talk about today is local development for LWC. We've heard from you all that you want a new version of the existing local dev that's been sitting around for a little bit. Now, with the new version of local dev, we've completely reimagined the architecture, so now we're sending your changes to the browser as you code, which makes you iterate faster and really visualize those changes in real time. Today for our task, we're working with coral clouds like everybody else, and we're taking a look at our contact page here. Within our contact page, we have our lightning web component where we can see what all the different memberships that our member has, and what we've been asked to do is actually add in an icon that will make it easier for our users to navigate to the associated record pages. And today we'll get to use localdev to help us move a little bit faster. So you all can see here that we've got localdev launched and ready to go from the CLI. And as I mentioned earlier, this new version of Localdev has been completely reimagined. And so what we're doing is we're sending your changes as you code to the browser via websocket, and then we can see those changes being reflected in the browser thanks to our hot module reloading. So we're going to go ahead and add in a little bit of space. We're going to add in our icon. And I want you all to pay really close attention, because as soon as Alba saves those changes, we're going to see them reflected in the browser. And there you have it. The changes are there and we didn't deploy. Super exciting. The changes are all there. We didn't deploy it. And you'll notice that those dropdowns we expanded earlier have stayed expanded. That's because not only do we support hot refreshes, but we're making sure that we're preserving your component state as well, which is super important to help you move just a little bit faster. Now, I've heard a rumor that coral cloud is using the same lightning web component for mobile as well. So let's take a look at what that looks like in mobile. Okay, I'm not a UX expert, I don't know if any of y'all are, but that's looking a little bit messy to me. So let's see if we can clean it up a little bit with the power of local dev for mobile. So again, you can see that we've got local dev launch and ready to go from the CLI. Alba's gonna go ahead and remove some of those extra membership details that we have. And again, make sure you're paying close attention, because as soon as we save these changes, you'll see it reflected in our mobile emulator. And there, uh, you almost have it. But luckily we can also show you some live debugging.

Speaker C **00:40:12**

Everything's real.

Speaker D **00:40:13**

It's all real, y'all. We're actually not using Figma for any of this. So while we're getting that loaded, the other really exciting announcement that we have is that not only is local dev available for mobile and web like we just saw, but it's also going to be available for experience cloud LWR sites, and we, uh, can clap for that, too.

Speaker A **00:40:41**

Cool.

Speaker D **00:40:42**

Yeah. And everything that we've been talking about across all three of these experiences for local dev are in beta and we can go ahead and check it out today in the CLI. Now we're going to save and we'll see those changes reflected in our mobile emulator. And there you have it. Kudos to Alba for the live debugging. Awesome. So we got a chance to see how we can move fast and debug fast with the power of Localdev. Now for our next, uh, assignment here, our coral cloud developers have also been asked to go ahead and actually take a look at all the different partner programs that our members are a part of. So it turns out that these corocloud members also have these other rewards programs that they can earn some membership points through and we need to go in and sum up all the points that they earn across all the programs. Now this would involve me working with a new apex controller and typically I'd have to do that all by myself. But today, very excited to announce that we're going to use agent force for developers to help us get started. Now, the great thing about agent force for developers is that it's actually built using our internal LLMs. And what that means is that we can have really tight feedback loops with the teams that are actually building these foundational models. So we can ensure that we're following the Salesforce best practices and we're staying up to date with them as they evolve three times a year. And second, we can also make sure that all of your code and your data are staying within our salesforce trust boundaries since we're not working with any of the external models here. Now, uh, that I'm feeling safe and ready to go with Agentforce, I'm going to actually ask our, uh, dev assistant to see if it can get me started with my apex method. And there you can see the dev assistant is thinking. And as we see that response stream in, let's pay attention to the fact that it's correctly relationally referencing our local metadata. So custom fields or custom objects that we may be working with, we're going to see those being referenced accurately. Now why is that? Well, because our team has been hard at work integrating retrieval, augmented generation, or Rag, into the dev assistant. So with Rag we're now able to understand your local metadata to make sure that we're seeing less of those hallucinations that you may have seen earlier. And of course I get a really helpful natural language text explanation alongside that generated code, which makes it a little bit easier for Alba and I to quickly read through it and decide whether or not we want to insert it into our editor. Now, today we've both read through it looks good, so we're going to add it to our editor and I'm taking a look at the rest of this file and I've realized I don't think Alva nor I wrote any of these methods, and I need to figure out what's going on here. Now, instead of me spending all my time now trying to read through the code, I'm going to ask our dev assistant to explain it for me. And just like that, with our helpful explain command, you get a nice natural language explanation of all the different methods in your class and what the purpose of that overall class is as well. So now that I've understood what this class is doing a little bit better, I've actually noticed that there is a method in here that's being used by our lightning web component, and I actually want to update the date format that we have in here. Now, instead of me or Alba figuring out how to do that, let's see if we can use the power of inline autocomplete to help us move a little bit faster. So you always have the option of manually triggering inline autocomplete if you'd like with our hotkeys. But today, let's see if Alba can start typing and we'll see that ghost text appear right there. And there you have it. We get our date format super quickly without us needing to look up any other syntax either. And if you get multiple completions, you as a developer always have the option of picking whichever one you want. And you have access to this in both Apex and LWC as well. Awesome. So now that we've gone ahead and accepted our changes, we're pretty much wrapped up, except for our very last step. We want to make sure that we're leaving this code maintainable for any future coral cloud developers. And so for that, I actually want to ask our dev assistant if it can help me out with actually documenting what these different methods in our class are. So with our slash document command, we're going to generate a new version of this class that has helpful apex comments going through each method describing what's going on. Super helpful, and just helps me move a little bit faster. Love it. Now that we've finished up all our changes, we should be ready to deploy this to the right. Almost, almost, y'all. We've got a few more things to do, but yes, we've got tests. We sadly won't get to that one today, but we've got tests. I do want you all to, uh, focus in on the fact that we were able to, to have our entire conversation history within the dev assistant as well, because with that we're able to ensure that we're just keeping that context going. And we really have the history of your previous prompts, making sure that your responses are a little bit more accurate. So let's pretend we've already written some tests, and now what we're going to do is make sure that the code that we've created is performant and scalable. And so for that, we got to use the power of scale center. So you can see here we've got scale center already activated inside our full copy sandbox. The great thing about Scale center is that it analyzes your code, it analyzes your test runs, and it, uh, even analyzes production events as recently as ten minutes ago. So our coral cloud developers who've been testing can go ahead, run those tests, explore the metrics, and then figure out where our spikes are. And we can see here today that we do have some spikes on our row lock errors and our cpu timeouts. So let's dig in a little bit deeper. We can go ahead and set a timeframe and then we can go and run an analysis based off of that. It does take a little bit of time, so we've already done this ahead of time. So let's jump over into our performance analysis tab. Within here, you can see all the recent analyses that we've run. We're going to go ahead and select that most recent one and it'll open up our consolidated report. Within this report, you get a really helpful view of all those different apex classes that may be contributing to your apex concurrency. And if we dig in a little bit deeper to our summary, you actually get a list of those apex classes that may be the slowest or the ones that are hitting those cpu timeouts. And we can see our offending class today. This would be the Coral cloud controller class. Someone tries saying that three times fast because it's hard. But we've got our Coral cloud controller class and I want to figure out what we can do to make it better. So for that, we're going to use the power of Apex guru. Apex Guru uses those same internal models we mentioned earlier to analyze your code and analyze your production and sandbox runtime profiling logs. This is super helpful because based off of that, we're understanding where those hotspots are and we can give you recommendations on how to address them. So let's take a look at the recommendation that ApexGuru is giving today. It has indeed caught that the Coral cloud controller class was using an inefficient global describe method. So we're going to go ahead and use those changes and we'll be able to deploy it over to the now, you might be wondering, why do I have to wait all the way until production to figure out where these hotspots are? And this has been a very common question. So we've been listening to your feedback and I'm excited to announce that we now have Apex guru integrated with code analyzer at development time. Now, uh, this is currently in dev, uh, preview as part of Codeanalyzer five. But it's great because as you all can see, we can actually scan our code at devtime and see whether or not we have those violations. In this case, we know that we've already addressed those violations, uh, in Apex guru itself. So we're good to go. And again, this feature is available to you with Codeanalyzer five, which is in dev, uh, preview right now. And we've actually completely reimagined this new version of Codeanalyzer as well. With this new version of code analyzer, you have the ability to go in and select exactly which rules you want to run across all of our different engines. And not only that, but you also have the ability to organize those rules by severity levels and tags, just making it cleaner and easier for you to work with. And lastly, you can customize those rules as well with our new YAML configuration file. So I know we saw a ton of different tools and features in action today with our work on coral clouds, but what I really want to point out to y'all is the fact that we were able to use all of these different tools as a part of our developer lifecycle. A big piece of feedback that we've heard from you all is that you want us to provide this functionality, but you want it all to be well integrated. And that's a key part of our vision moving forward. So I hope you all enjoyed learning a little bit more about how you can build more productively and efficiently on the platform today. And with that, hand it back to you, Christoph.

Speaker A 00:50:34

All right, thank you so much, Anania, Stefan J and Alba. This was incredible content. We saw how to build so much with data, AI and tools. And of course, to learn more, we created that developer quest. Take your phones out and scan that QR code that is packed with learnings to learn more about this and hands on opportunities. And with that I want to thank you all, uh, for attending today, and safe travels back home. Thank you so much.