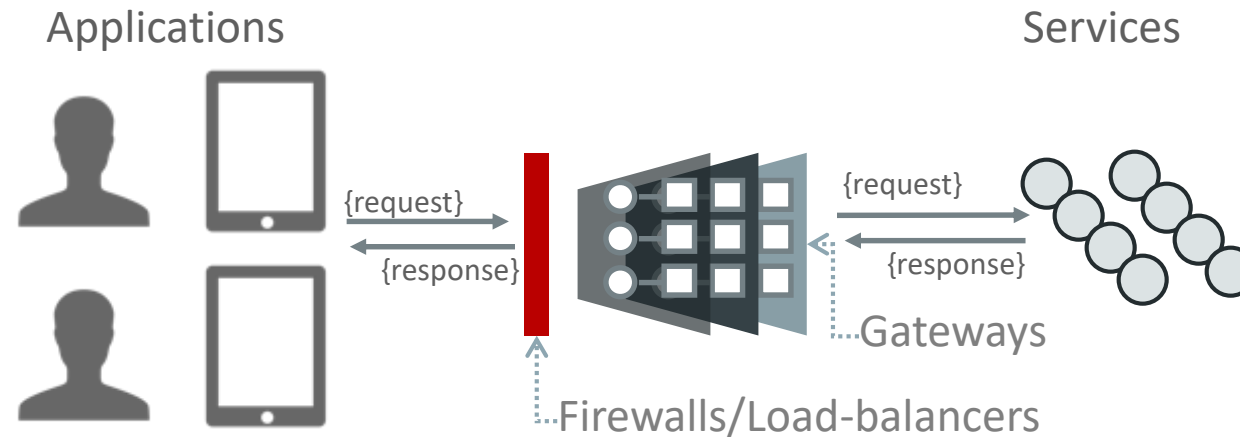


APIs are the Doors to Digital Transformation

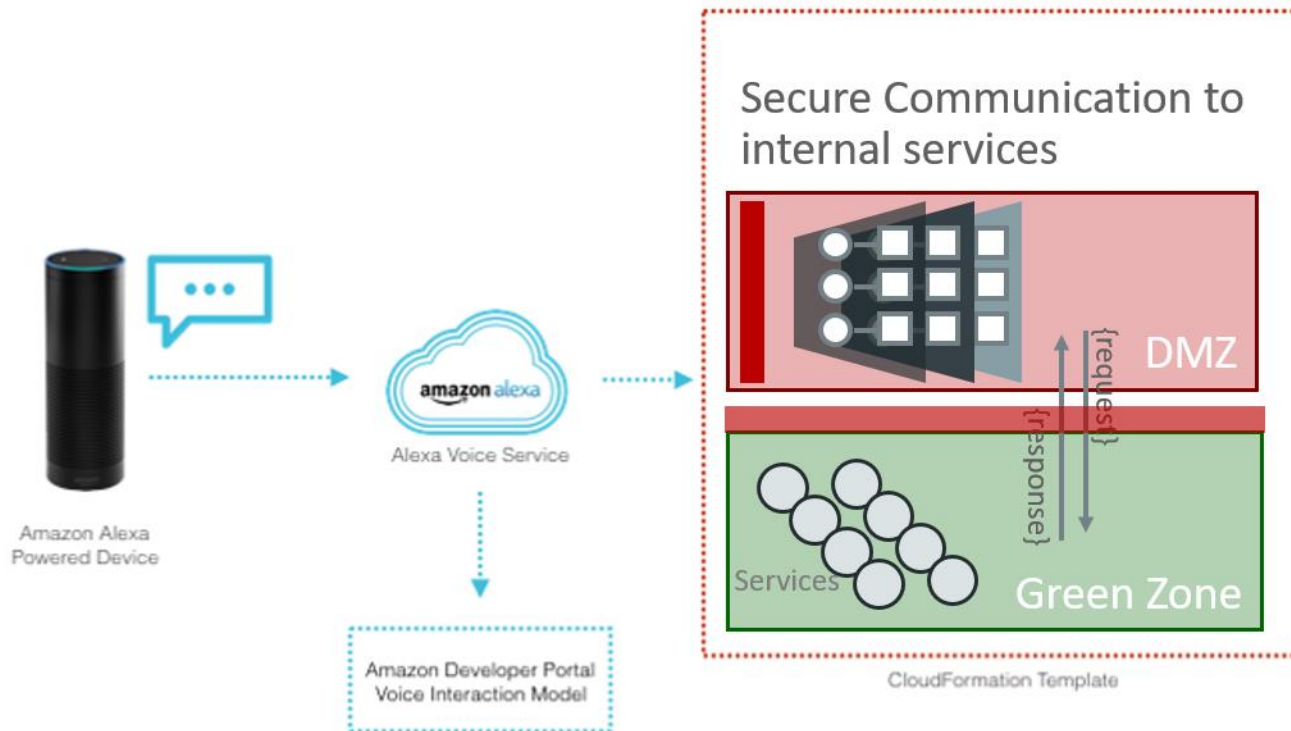
Secure, and Control Access to Services



Lot of our customers running an API first strategy. This mean first the logic and then the User Interface. A user interface could be everything nowadays: A device, an app, a browser or flat client.

Real use case

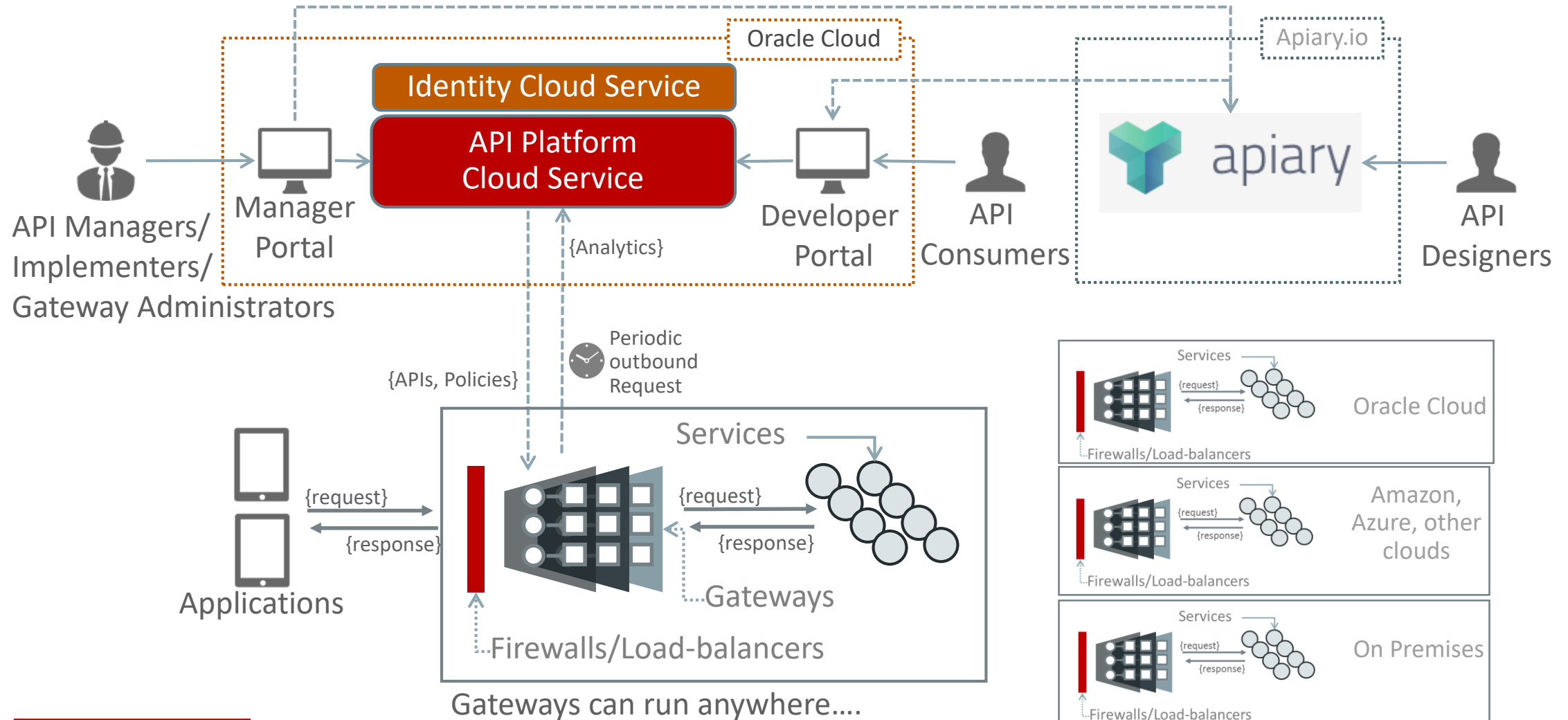
Audio assistant Alexa



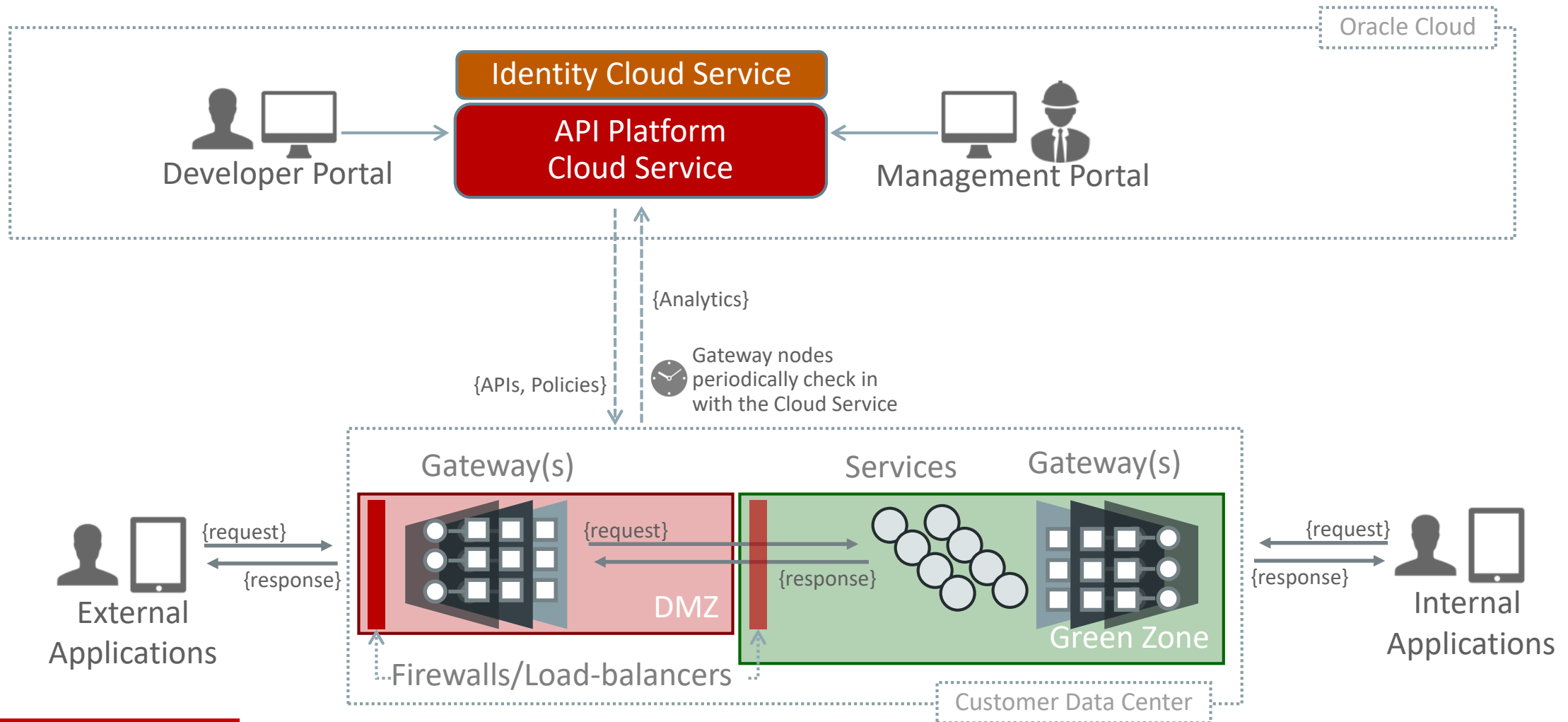
Use cases

- Automatic assistant e.g. for a reception desk...
- Audio information what ever you can imagine
 - Business summary a la „Alexa what are my business news“
„Alexa: Won deals 5 Mio US\$,“

Architecture of API Platform Cloud Service



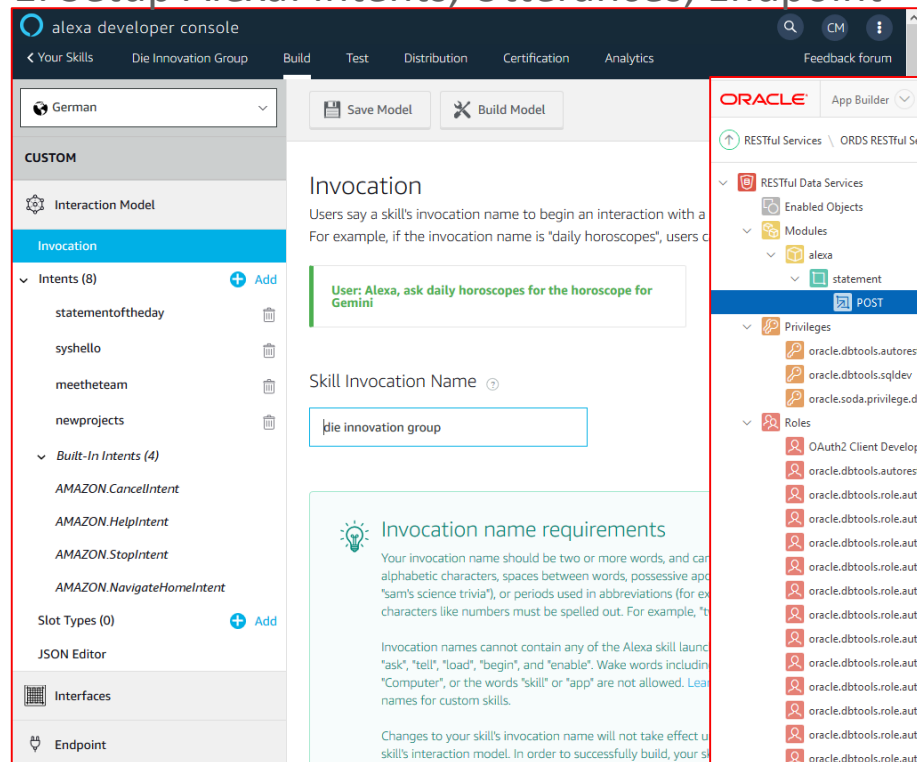
Creating On-Premise APIs for On-Premise Services for Both External and Internal Clients



Make it the short way

How to implement Alexa APP very fast

1. Setup Alexa: Intents, Utterances, Endpoint



alexaskill developer console

German

Save Model Build Model

CUSTOM

Interaction Model

Invocation

Intents (8)

- statementoftheday
- syshello
- meetheteam
- newprojects

Built-In Intents (4)

- AMAZON.CancelIntent
- AMAZON.HelpIntent
- AMAZON.StopIntent
- AMAZON.NavigateHomeIntent

Slot Types (0)

JSON Editor

Interfaces

Endpoint

Invocation

Users say a skill's invocation name to begin an interaction with a skill. For example, if the invocation name is "daily horoscopes", users can say "Alexa, ask daily horoscopes for the horoscope for Gemini".

Skill Invocation Name

die innovation group

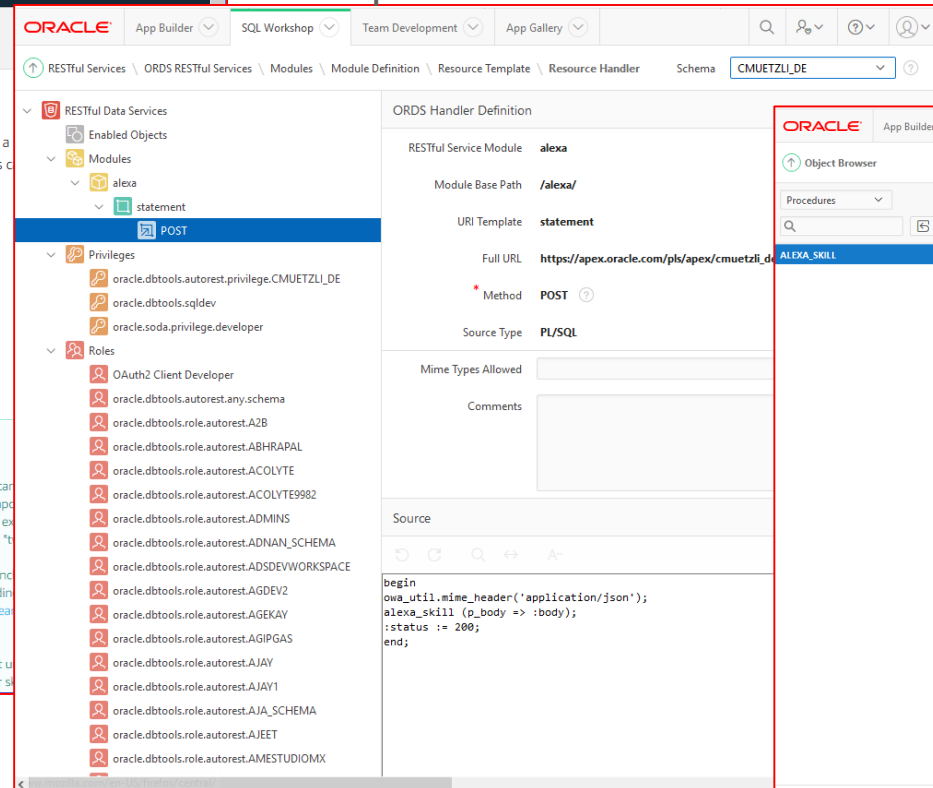
Invocation name requirements

Your invocation name should be two or more words, and can contain alphabetic characters, spaces between words, possessive apostrophes (e.g., "sam's science trivia"), or periods used in abbreviations (for example, "Dr."). Characters like numbers must be spelled out. For example, "one two three".

Invocation names cannot contain any of the Alexa skill launch words: "ask", "tell", "load", "begin", and "enable". Wake words including "Computer", or the words "skill" or "app" are not allowed. Learn more about naming conventions for custom skills.

Changes to your skill's invocation name will not take effect until you rebuild your skill's interaction model. In order to successfully build, your skill's invocation name must be unique.

2. Setup the API



ORACLE App Builder SQL Workshop Team Development App Gallery

RESTful Services ORDS RESTful Services Modules Module Definition Resource Template Resource Handler Schema CMUETZLI_DE

RESTful Data Services

- Enabled Objects
- Modules
 - alexaskill
 - statement
- Privileges
 - oracle.dbtools.aotrest.privilege.CMUETZLI_DE
 - oracle.dbtools.sqldev
 - oracle.soda.privilege.developer
- Roles
 - OAuth2 Client Developer
 - oracle.dbtools.aotrest.any.schema
 - oracle.dbtools.role.aotrest.A2B
 - oracle.dbtools.role.aotrest.ABHRAPAL
 - oracle.dbtools.role.aotrest.ACOLYTE
 - oracle.dbtools.role.aotrest.ACOLYTE9982
 - oracle.dbtools.role.aotrest.ADMINS
 - oracle.dbtools.role.aotrest.ADNAN_SCHEMA
 - oracle.dbtools.role.aotrest.ADSDEVWORKSPACE
 - oracle.dbtools.role.aotrest.AGDEV2
 - oracle.dbtools.role.aotrest.AGEKAY
 - oracle.dbtools.role.aotrest.AGIPGAS
 - oracle.dbtools.role.aotrest.AJAY
 - oracle.dbtools.role.aotrest.AJAY1
 - oracle.dbtools.role.aotrest.AJA_SCHEMA
 - oracle.dbtools.role.aotrest.AJEET
 - oracle.dbtools.role.aotrest.AMESTUDIOMX

ORDS Handler Definition

RESTful Service Module alexaskill

Module Base Path /alexaskill/

URI Template statement

Full URL https://apex.oracle.com/pls/apex/cmuetzli.de/alexaskill

Method POST

Source Type PL/SQL

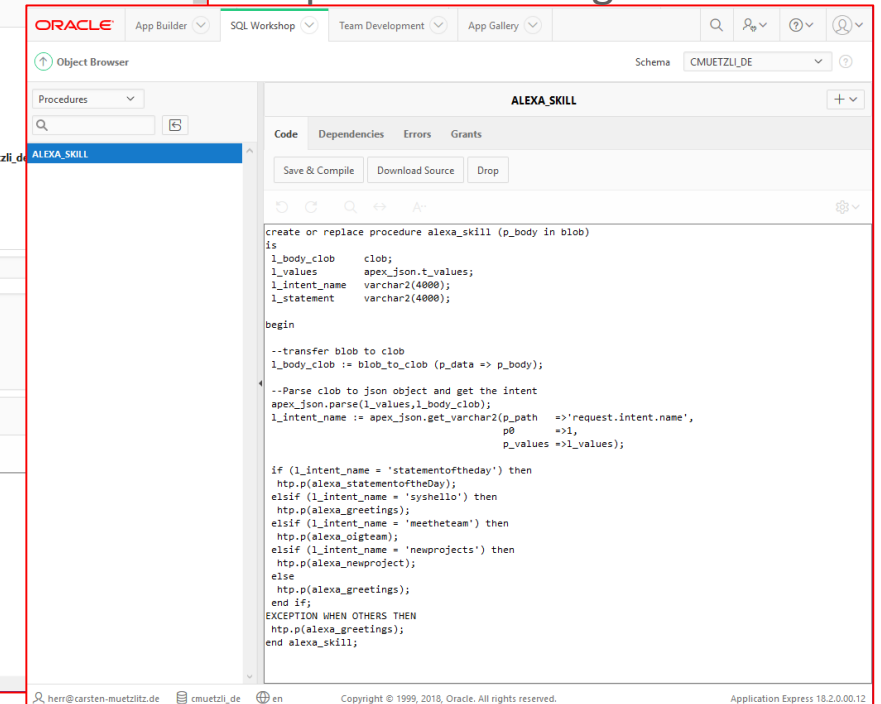
Mime Types Allowed

Comments

Source

```
begin
owa_util.mime_header('application/json');
alexaskill(p_body => :body);
:status := 200;
end;
```

3. Implement the logic



ORACLE App Builder SQL Workshop Team Development App Gallery

Object Browser

Procedures

ALEXASKILL

Code Dependencies Errors Grants

Save & Compile Download Source Drop

```
create or replace procedure alexaskill (p_body in blob)
is
l_body_clob clob;
l_values apex_json.t_values;
l_intent_name varchar2(4000);
l_statement varchar2(4000);
begin
--transfer blob to clob
l_body_clob := blob_to_clob (p_data => p_body);

--Parse clob to json object and get the intent
apex_json.parse(l_values,l_body_clob);
l_intent_name := apex_json.get_varchar2(p_path => 'request.intent.name',
p0 => l_values);

if (l_intent_name = 'statementoftheday') then
http.p(alexaskill_statementoftheday);
elseif (l_intent_name = 'syshello') then
http.p(alexaskill_greetings);
elseif (l_intent_name = 'meetheteam') then
http.p(alexaskill_oigteam);
elseif (l_intent_name = 'newprojects') then
http.p(alexaskill_newproject);
else
http.p(alexaskill_greetings);
end if;
EXCEPTION WHEN OTHERS THEN
http.p(alexaskill_greetings);
end alexaskill;
```

How Alexa is working

Like a chatbot

- You will create an invocation name like „Die Innovation Group“
 - This guy will open your skill within Alexa Device
- You will create intents like „meettheteam“
 - Then you will write down typical description sentences so called utterances, e.g.
 - „Who is the innovation group“
- Then you enter an API Endpoint with the logic
 - If you talk to Alexa, she will try to response with the right intent
 - This intent will be send to API
 - API will answer with the right information for that specific intent

It is really very easy to understand. You need only 3 simple pieces. The main part is really developing th API.