Unit 5 – Build SAP Extension with SAP AppGyver

Certification: C_LCNC_01

Low-Code / No-Code Applications

Appointments Application with SAP AppGyver

Demo: Appointments application

SAP Development Learning

Features

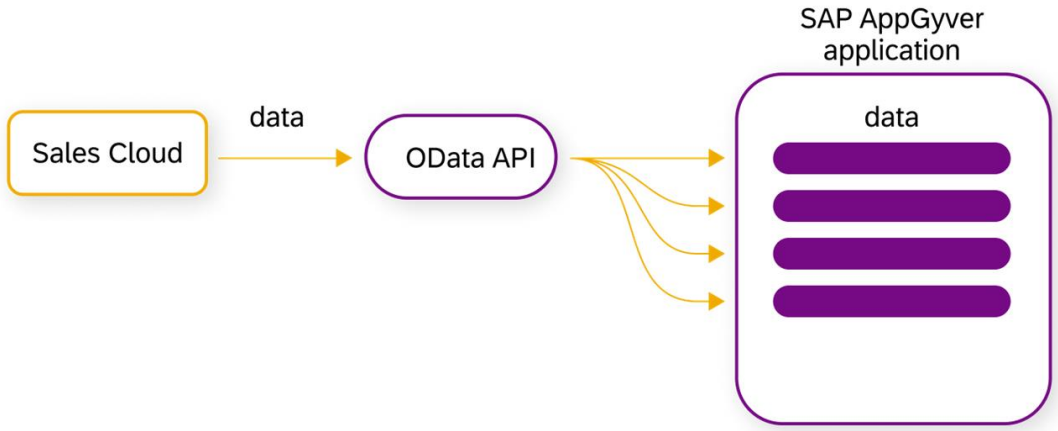
- Cross platform – Mobile (iOS, Android)
- OData API Integration – SAP Sales Cloud
- Composite Component
- More complex binding

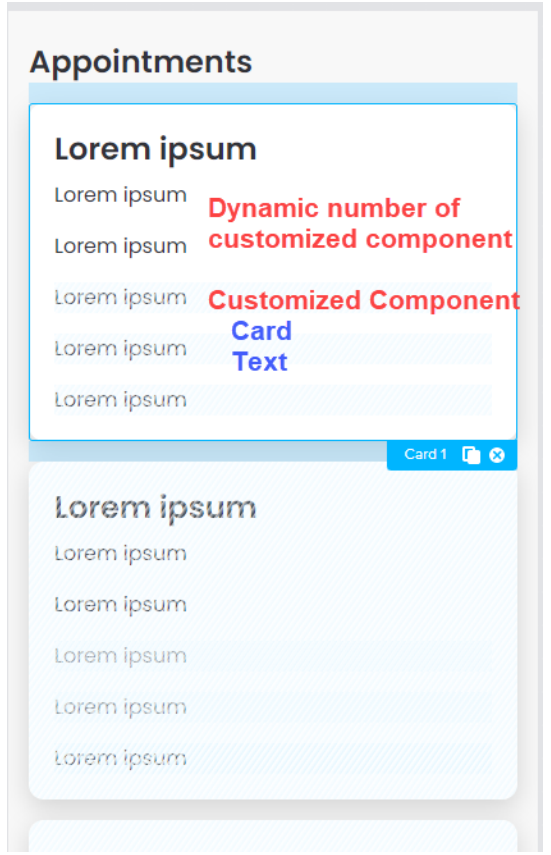
Why SAP Extensions with SAP AppGyver ?

- SaaS solutions from SAP cover a majority of customer use cases
- However, every customer has unique needs that cannot be covered out-of-the-box
- Various options to extend SAP solutions (In-App, Side-by-Side)

High level plan for UI, Logic and Data

- Build application to display appointments
- Make connection to SAP Sales Cloud OData API
- Display appointments data





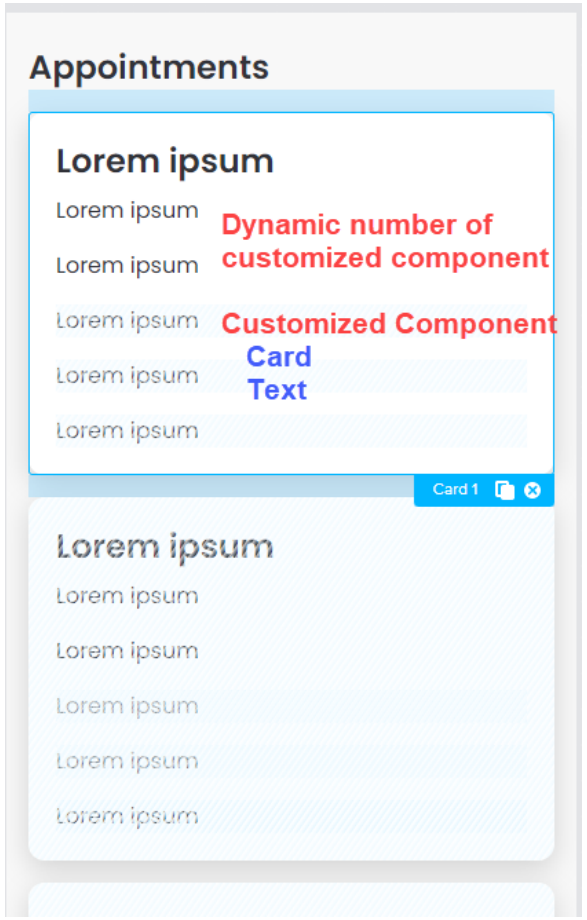
Steps involved

1. Setting up the pre-requisites
2. Setting up the UI – Single View and Components
3. Connections to OData API
4. Binding, displaying data in UI

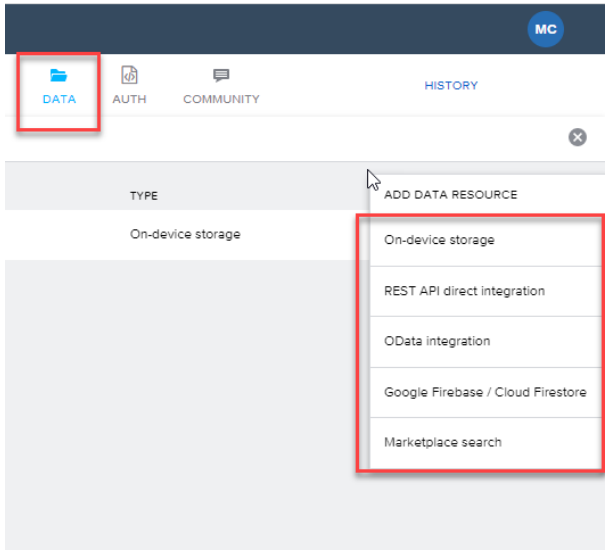
Step 1 – Pre-Requisites

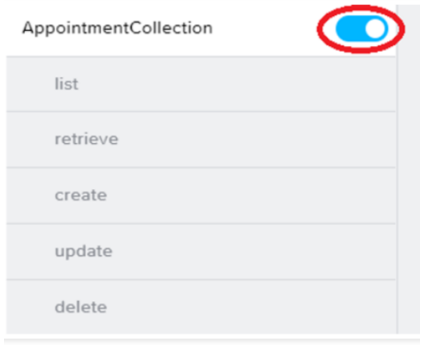
1. Subscribe to SAP AppGyver via a booster from SAP Business Technology Platform
2. Get the SAP AppGyver Preview app on your mobile phone

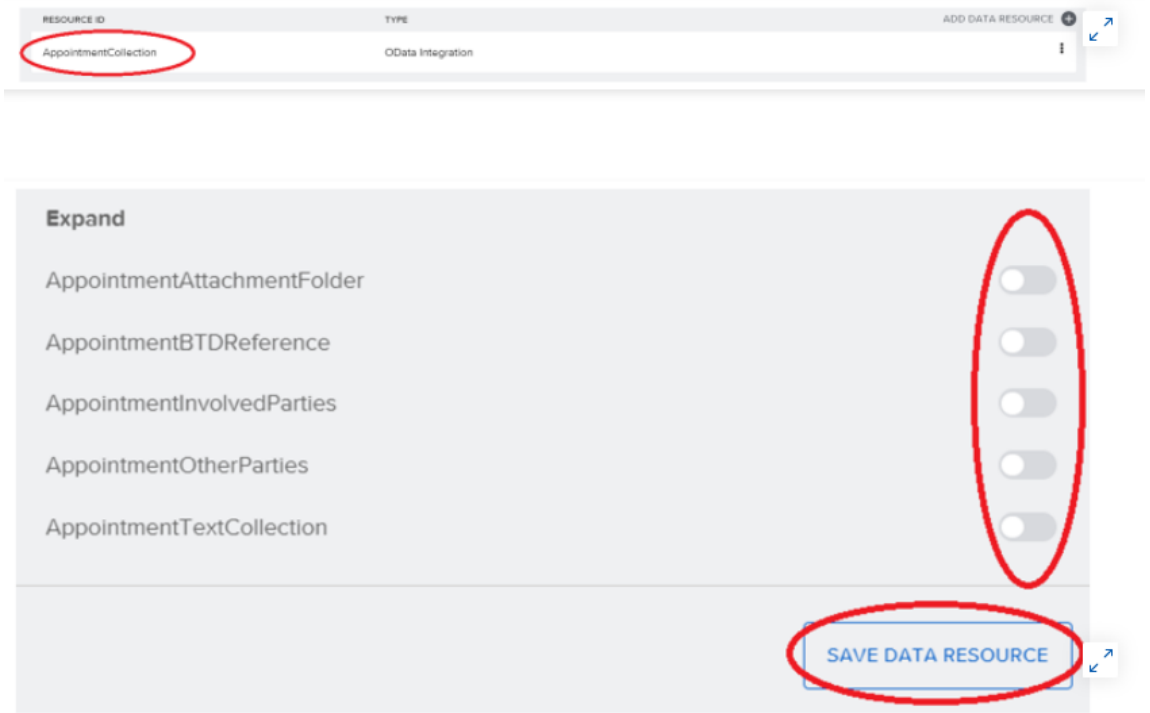
Step 2 – View and its components



Step 3 – Connections to OData API







Step 3 – Connections to OData API

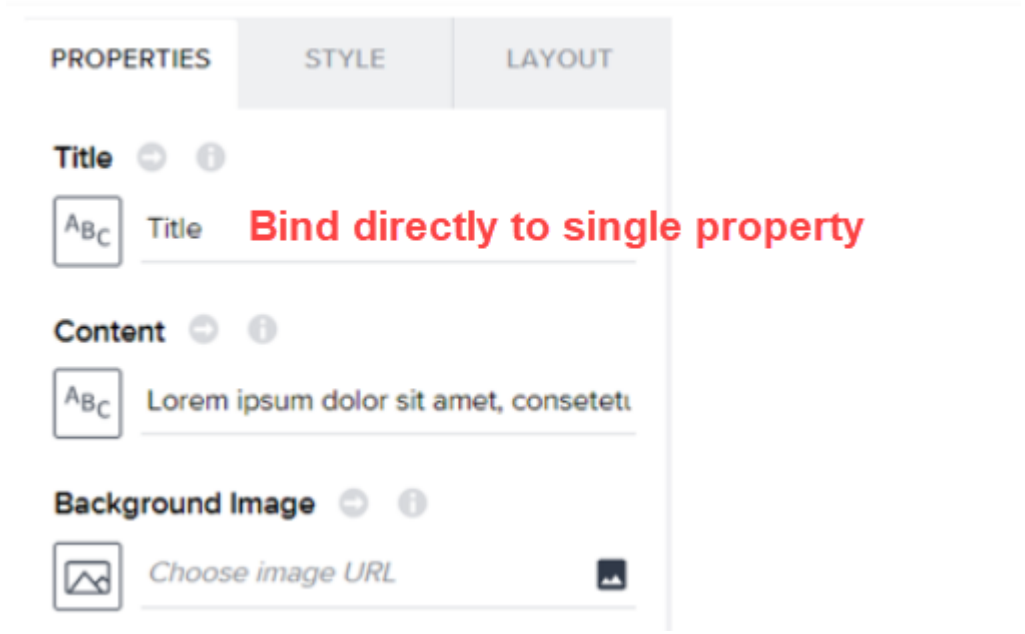
https://my346234.crm.ondemand.com/sap/c4c/odata/v1/c4codataapi/$metadata

Credentials

Demouser / Welcome1

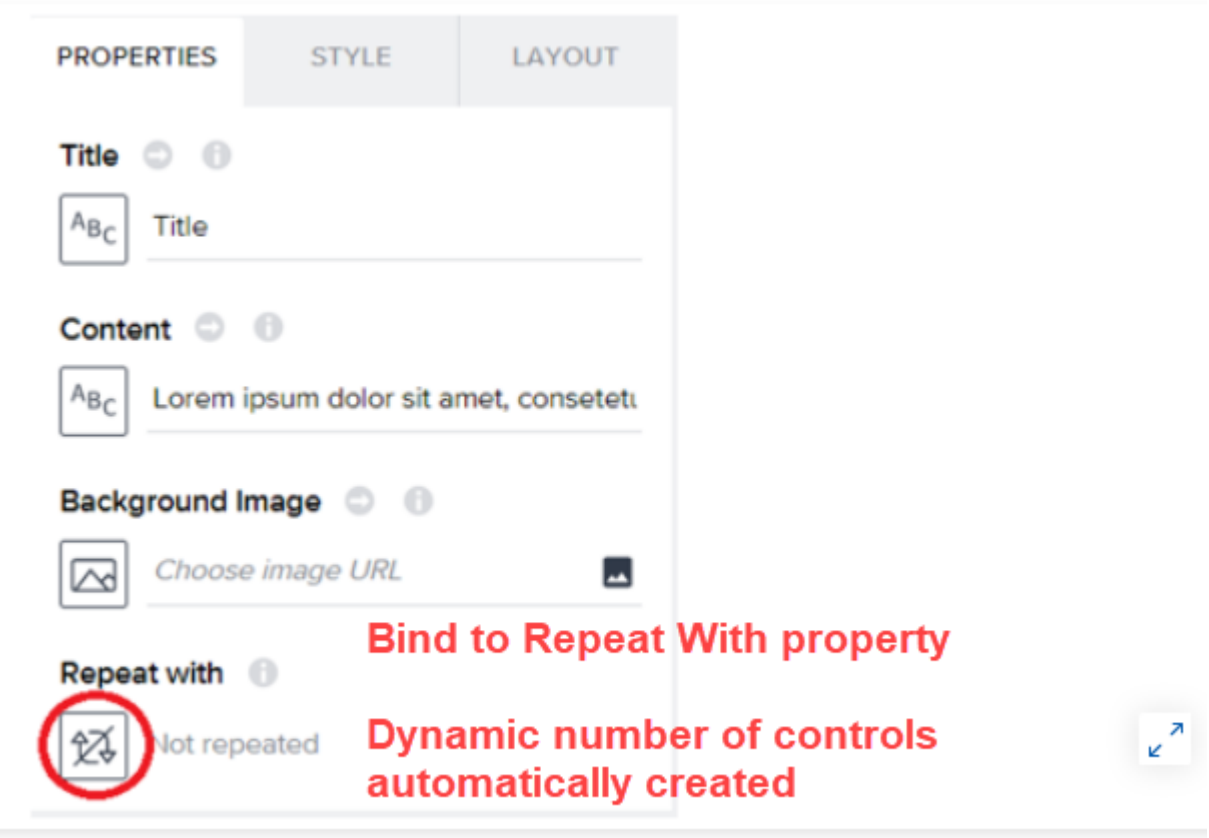
Step 4 – Binding, displaying data in UI

```
{
    "name": "Milton"
}
```



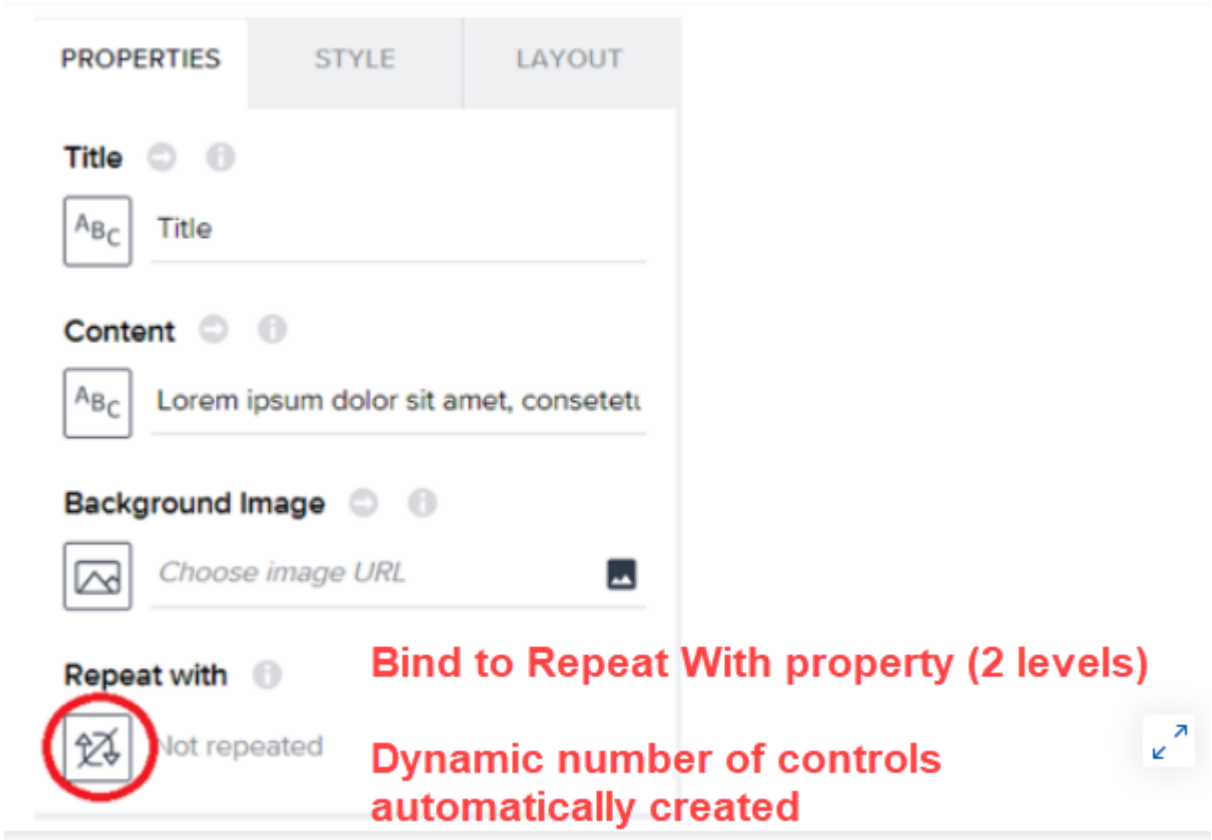
Step 4 – Binding, displaying data in UI

```
[
   {
      "name": "Milton"
   },
   {
      "name": "Peter"
   },
   {
      "name": "Pavel"
   },
   {
      "name": "Jose"
   }
]
```



Step 4 – Binding, displaying data in UI

```
[
   {
      "name": "Milton",
      "hobbies": ["tennis", "chess", "soccer"]
   },
   {
      "name": "Peter",
      "hobbies": ["hiking", "cooking"]
   },
   {
      "name": "Pavel",
      "hobbies": ["tennis", "rugby"]
   },
   {
      "name": "Jose",
      "hobbies": ["soccer", "athletics"]
   }
]
```



What we have learned so far

- SAP Sales Cloud
- OData API Integration
- Composite Components
- Multi-level binding

Key Summary Points – Unit 5

Q10. What can SAP Sales Cloud assist with?

- ✓ communicating and collaborating with customers
- ✓ supporting in the lead-to-cash process
- C creating applications
- ✓ giving complete transparency where you stand with current sales business
- E managing your employees

✓ Correct

Correct. SAP Cloud can assist you with giving complete transparency where you stand with current sales business, supporting in the lead-to-cash process and, communicating and collaborating with customers.

Key Summary Points – Unit 5

Q9. How did we open the Component Template Editor?

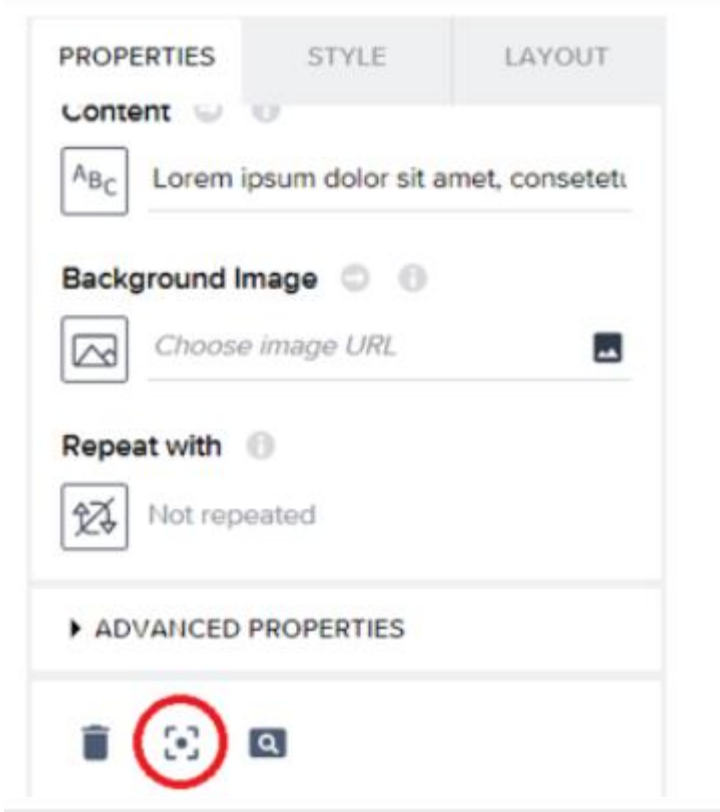
A Dragging card components into the canvas

Clicking on the Component then Scrolling down and clicking on the square with the dot in the middle

C Double clicking on the outside of the canvas

Correct

Correct. We open the Component Template Editor by clicking on the Component then Scrolling down and clicking on the square with the dot in the middle.



Key Summary Points – Unit 5

Q7. What is the use for APIs?

A connect with colleagues

✓ perform some action on the system

C create applications

✓ get data from the system

✓ communicate with a system

✓ Correct

Correct. The use of APIs are to communicate with a system, get data from the system and perform some action on the system.

Key Summary Points – Unit 5

Q2. What do we see when we open the URL in the browser?

A The data stored on your PC

B The SAP AppGyver internal data

✓ The XML metadata of the OData service

✓ Correct

Correct. When we open the URL in the browser, we can see the XML metadata of the OData service.

Key Summary Points – Unit 5

```
[
    {
        "name": "Milton"
    },
    {
        "name": "Peter"
    },
    {
        "name": "Pavel"
    },
    {
        "name": "Jose"
    }
]
```

Q5. How can we know that the data is a list of objects?

A We can open it in Google

First wrapped in brackets [] and then in curly brackets {}

C Since it´s the XML metadata of the OData service

Correct

Correct. We know that it is a list of objects, if the data is first wrapped in brackets [] and then in curly brackets {}.