# WSO2 Con 2024 - Choreo Tutorial

## Prerequisites

1. GitHub account
2. VSCode

## Section 1: Deploying a Microservice in Choreo

### Create the database

1. Login to Choreo (https://console.choreo.dev/)
2. Go to the Organization home page
3. From the left menu select Dependencies → Databases
4. Use the following data to create a database server

| Field | Value |
|---|---|
| **Select Storage** | MySQL |
| **Service Name** | appointmentsdb |
| **Select Cloud Provider** | Digital Ocean |
| **Select Region** | United States |
| **Select Service Plan** | Hobbyist |

5. Click on the "Create" Button

### Create the service component

1. Fork https://github.com/wso2con2024/choreo-tutorial
2. Clone the forked repo to your workstation

```
Unset

                git clone <REPO_URL>
```

3. Open a new browser tab and go to https://console.choreo.dev/ and Sign in

4. Go to the Organization home page
5. Create two projects with following names:
    a. Web Portal
    b. Appointment Management
6. Go to "Appointment Management" Project
7. Click on the **"Service"** card to create new service component

| Field | Value |
|---|---|
| **Component Name** | Appointment Service |
| **Connect Your Repository** | <YOUR_FORKED_REPO> |
| **Buildpack** | NodeJS |
| **NodeJS Project Directory** | /appointment-management/appointment-service |
| **Language Version** | 20.x.x |

6. Click on "**Appointment Service**"
7. Select **Build** in left menu
8. Click **"Build Latest"** button
9. Go to "**Deploy**" and click on "**Configure & Deploy**" button
10. Use the environment configuration pane to configure following environment variables. These values need to be copied from the database we created earlier. (Please navigate to to Organization → Dependencies → Databases → appointmentdb to find corresponding values)
    a. DB_HOST
    b. DB_PORT
    c. DB_NAME
    d. DB_USER
    e. DB_PASSWORD
11. Complete the wizard and click on "**Deploy**" button
12. Go to Test → Console
13. Click on the "**Generate URL**" button.
14. Test the API by creating a new appointment. (I.e. You can send a POST request to /appointments)
15. Go to Deploy page and click on "**Promote**" button in the Development card
16. From the configuration type pane select "**Use Development configuration**". Complete the wizard.

# Section 2: Service to Service Communication

1. Go to "Web Portal" project
2. Create a service component by clicking on the Service card. Use following details

| Field | Value |
|-------|-------|
| Component Name | Backend for frontend |
| Connect Your Repository | <YOUR_FORKED_REPO> |
| Buildpack | NodeJS |
| NodeJS Project Directory | /web-portal/backend |
| Language Version | 20.x.x |

3. Navigate to the newly created component (i.e. Backend for frontend)
4. Go to Dependencies → Connections and click on "**Create**" button
5. Select the "Appointment service" card and give this connection a name ("Appointment connection") in the subsequent screen.
6. Click on the **Create** button
7. From the subsequent UI, copy the generated connection configuration to

   `YOUR_FORKED_REPO/web-portal/backend/.choreo/component-config.yaml`

Your changed component-config.yaml may look like this:

```
21          # The path should be relative to the Docker context.
22          schemaFilePath: openapi.yaml
23      outbound:
24        serviceReferences:
25        - name: choreo:///choreotutorial2024/appointment-management/siffyq/8b1c3/v1.0/ORGANIZATION
26          connectionConfig: 01ef0609-4a22-12a6-9dab-f799362d4a37
27          env:
28          - from: ServiceURL
29            to: APPOINTMENT_SERVICE_URL
30          - from: ConsumerKey
31            to: APPOINTMENTS_OAUTH_CLIENT_ID
32          - from: ConsumerSecret
33            to: APPOINTMENTS_OAUTH_CLIENT_SECRET
34          - from: TokenURL
35            to: APPOINTMENTS_OAUTH_TOKEN_URL
36
```

8. Commit the change and push to GitHub repo

```
Unset
    git add .
    git commit -m "Add connection configurations"
    git push origin
```

9. Go to the **Build** page and use the "**Show commits**" option to select the latest commit and trigger a build.

10. Go to the **Deploy** page and click on "**Configure & Deploy**". Complete the wizard.

11. Go to **Test → Console** and expand the GET resource. Try out the API following data.

| Field | Value |
|-------|-------|
| **email** | johndoe@example.com |

12. Go to the **Deploy** page and promote the service to the "**Production**" environment. Complete the wizard with default settings.

13. Test the production environment to verify.

# Section 3: Deploying  a web application on Choreo

## Create the web application component

1. Make sure you are on "Web Portal" project overview page
2. Click on the "**Create**" button and select the **Web Application** card. Use following details to create the component.

| Field | Value |
|---|---|
| **Component Name** | Frontend |
| **Connect Your Repository** | <YOUR_FORKED_REPO> |
| **Buildpack** | React |
| **Project Directory** | /web-portal/frontend |
| **Build Command** | npm run build |
| **Build Path** | /build |
| **Node Version** | 20 |

3. Go to **Dependencies → Connections** and click on the **Create** button

4. Select "Backend for frontend" card and give this connection a name ("backend connection)

5. Copy the connection configuration value for use in deployment. This is necessary because we're using Choreo-managed authentication to authenticate with the backend.
   For example:

```
Unset
window.configs = {
    apiUrl: '<<value from your connection>>',
};
```

6. Go to the "**Build**" page and click on the "**Build Latest**" button.

7. Navigate to the "**Deploy**" page and select the "**Configure & Deploy**" button. Enter the value you copied earlier into the "Configure & Deploy" pane. It should appear as shown in the screenshot below:

8. Complete the wizard with default settings

## Configure the Choreo Identity Provider

1. Go to Organization overview page
2. Go to **Settings → Application Security**.
3. Click on the **Manage** link on "**Choreo Built-in Identity Provider**" card
4. Select the file `YOUR_FORKED_REPO/web-portal/frontend/userstore.csv` and click the "**Re-upload**" button to update the Choreo Identity Provider.
5. Navigate back to **"Web Portal" → Frontend** component. Then go to the **Deploy** page and click on the **Web App URL** in the development card.
6. Use following credentials to login and tryout the application

| Field | Value |
|---|---|
| **Username** | john |
| **Password** | user@1234 |

## Configure Asgardeo as Identity Provider in production

1. Make sure you are on **"Web Portal" → Frontend** in the browser.
2. Go to Component settings by clicking on the "**Settings**" → **Authentication Keys**. Select "**Production**" tab
3. Select "**Asgardeo**" as the identity provider.
4. Expand **OIDC App Configuration** pane. We need to copy the Redirect URLs to configure Asgardeo in later step

5. Go to https://console.asgardeo.io. Sign in to Asgardeo with the same credentials
6. Make sure you have selected the same organization you used in Choreo
7. Go to **"User Management"→Users** and click on the **"Add User→Single User"** button.
8. Use Following details to add a user

| Field | Value |
|-------|-------|
| **Username (Email)** | <A VALID EMAIL ADDRESS> |
| **Select the method to set the user password** | Set a password for the user |
| **Password** | Demo@1234 |

9. Go to "**Applications**" in the left menu
10. Click the "**New Application**" button
11. Select Standard-Based Application card
12. Use the following details and click on the "**Create**" button

| Field | Value |
|-------|-------|
| **Name** | CareConnect Web Portal |
| **Protocol** | OAuth2.0 OpenID Connect |

13. Make sure you are on the protocol tab of the newly created application
14. Fill in the following details and press the "**Update**" button.

| Field | Value |
|-------|-------|
| **Allowed grant types** | Code, Refresh Token |
| **Authorised Redirect URLs** | <Redirect URLs copied from Choreo console's OIDC App Configuration> |
| **Allowed origins** | <Domain URL extracted from the "Authorised Redirect URLs"> E.g. https://6b3d2f4e-77d4-4299-b386-3ffc86f11fa2.e1-us-east-azure.choreoapps.dev |
| **Access Token** | JWT |

15. Go to the **User Attributes** tab. Mark following user attributes as mandatory and click on "Update"
    a. Email→ Email
    b. Profile→ Full Name

16. Go to the **Protocol** tab and Copy "Client ID" and "Client Secret" to Choreo console's OIDC App Configuration. Click on the "**Add Keys**" button afterwards.

17. Go to the Deploy page and click the **Promote** button. Complete the wizard with "Use Development configurations".

18. Navigate to "**Web App URL**" in the production card. Log into the web application using the user added in the asgardeo.

## Section 4: Versioning and Deployment Tracks

We are going to develop a feature in Appointment Service to support Dental Checkups. Required code changes are already committed to the git branch "support_for_dental_checkups". In this session we will ship this feature to production.

1. Make sure you are on **Appointment Management** → **Appointment Service** in the browser.

2. Click on **Deployment Track** drop down. Click on "Create New" menu item.

3. Use following information to fill the create deployment track dialog box.

| Field | Value |
|---|---|
| **Branch Name** | support_for_dental_checkups |
| **API Version** | v1.1 |

4. Go to build and then click on **Build Latest** button

5. Go to "**Deploy**" and click on "**Configure & Deploy**" button

6. Use the environment configuration pane to configure following environment variables. These values need to be copied from the database we created earlier. (Please navigate to to Organization → Dependencies → Databases → appointmentdb to find corresponding values)
    a. DB_HOST
    b. DB_PORT
    c. DB_NAME
    d. DB_USER

e. DB_PASSWORD
7. Complete the wizard and click on the "**Deploy**" button.
8. Go to Test → Console
9. Click on the "**Generate URL**" button.
10. Test the API by retrieving appointment types. (I.e. You can send a GET request to /appointment-types)
11. Go to Deploy page and click on "**Promote**" button in the Development card
12. From the configuration type pane select "**Use Development configuration**". Complete the wizard.
13. Now that the feature is in production Web application should automatically show the "Dental Check up" in GUI.
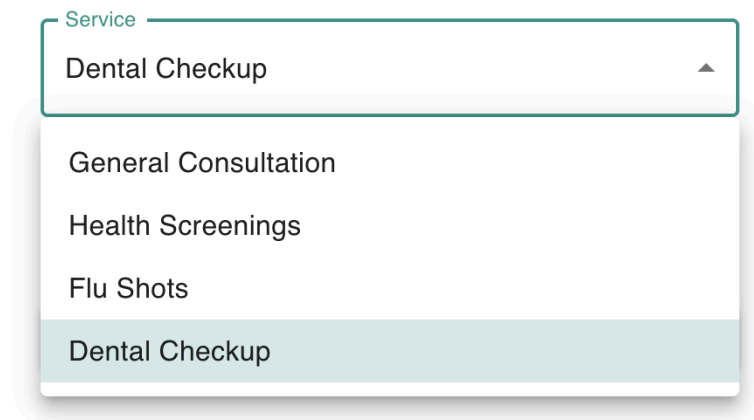


# Section 5: Scheduling a Task

1. Make sure you are on **Appointment Management** project
2. Click on **Create** button and select **Scheduled Task** card
3. Use following details to create the component.

| Field | Value |
|---|---|
| **Component Name** | Email Reminder Task |
| **Connect Your Repository** | <YOUR_FORKED_REPO> |
| **Buildpack** | Ballerina |
| **Project Directory** | /appointment-management/email-reminder-task |

4. Go to the "**Build**" page and click on the "**Build Latest**" button.

5. Go to **Appointment Management → Appointment Service** overview page and copy the Project URL
6. Navigate to the **Appointment Management → Email Reminder Task → Deploy** page and select the "**Configure & Deploy**" button.
7. Put the copied value to **appointmentApiUrl** and click **Next**
8. Use following values in the **Schedule** pane and click **Deploy**

| Field | Value |
|---|---|
| **Select Time Zone** | Default Value |
| **Select Range** | Day |
| **Every** | 1 |
| **At** | 01:00 AM |

9. Go to **Execution** page and click on **Run Now** button.