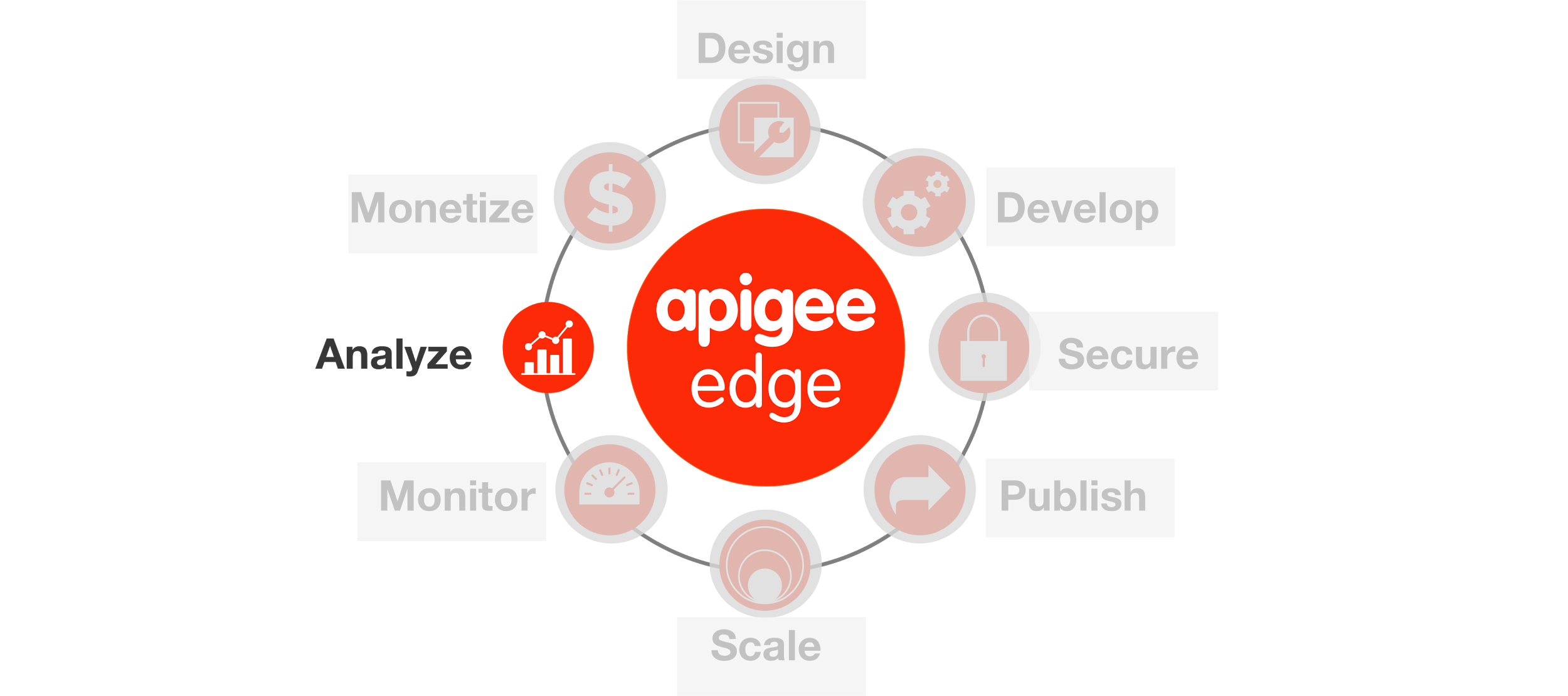


**Lab 3 - Create Developer portal**



**Overview**

The **Developer Services** component is a ready built **developer portal** that is based on Drupal. The portal comes out-of-the-box with community features such as features like Blogs, Forums, FAQs etc that help build a great developer ecosystem either internally within Walmart devlopers and/or externally exposed to partner and third party devlopers. It is easily customizable (can be easily skinned with Walmart colors, fonts, branding, etc.). The portal includes a Smart Key Management application for **secure registration and onboarding** of developers (whether internal, partner, or external).

**Objectives**

The objective of this lesson is to get you familiar with Apigee’s Developer Portal.

**Prerequisites**

* A dev portal is provisioned

**Estimated Time: 45 mins**

Now we will enhance the proxy to start collecting some statistics from the response payload. As part of this we will add a couple of policies including a Statistics Collector to the existing proxy that we have just created. Lets add a policy **–**

**To add a user from the developer portal:**

1. The user selects **Register** on the developer portal home page.
2. The registration page appears.
3. The user enters the required information and selects **Create new account**.  
   Depending on the new account registration settings, when the new account is created, the user is sent an automated welcome email.

**To register an app through the default developer portal:**

1. Select **My apps** below your username in the login menu.
2. Click the **Add  (+)**  new app  icon.

## 

dd app icon

Enter appropriate values in the Add app page:

|  |  |
| --- | --- |
| **App name** | Provide a unique App Name for your app. |
| **Callback URL** | If required, provide the callback URL to which the OAuth call should return.  By default, this value is required even if your APIs do not use OAuth. If your APIs do not use OAuth, enter a dummy value such as my.site.com. |
| **Product** | Select one or more API products. |

6. Click C**reate App** to register the app and generate the API key for the app.

**To enable analytics on the portal:**

1. Log in to your portal as a user with admin or content creation privileges.
2. Select **Configuration > Dev Portal Settings >  Dev Portal App Config** in the Drupal administration menu.
3. On the Dev Portal App Config page, expand the **App Performance** settings area.
4. Select the check box for "Show Developer App performance tab?".
5. Select either **Test** or **Production** for the "Environment to query for analytics data". For a publicly available portal, you will most likely select **Production**.
6. Save the configuration

**Create documentation in the portal**

**To create a model**

When you create a model, it's stored in your Edge organization as the source for the API structure. For more information, see [About SmartDoc models and templates](http://apigee.com/docs/developer-services/content/using-smartdocs-document-apis#models).

1. Log in to your portal as a user with admin or content creation privileges.
2. Select **Content > SmartDocs** in the Drupal administration menu.
3. Select **New model** at the top of the page.
4. Enter the following fields:
   * **Name**: The model name that will be displayed across the site.
   * **Internal name**: As you type the **Name**, the internal name displays. The internal name for the model that must be unique among all models.  The internal name must contain only lowercase letters, numbers, and hyphens with no spaces. Select**Edit** to edit this name.
   * **Description**: A description of the model.
5. Select **Create Model**.

**Import a Swagger API description**

After you have successfully created a model, you can import a Swagger API description. Edge supports Swagger version 1.2 and 2.0.

Swagger uses files containing JSON objects to describe an API. Every time you import a Swagger definition, you automatically create a new revision of the model.

Before you import your Swagger file, you can use tools to validate it. For example, [swagger-tools](https://www.npmjs.com/package/swagger-tools) provides tools to validate a Swagger document. Or, you can pass the URL of your Swagger file to this URL to validate it:

http://online.swagger.io/validator/?url=***swaggerURL***

Replace ***swaggerURL*** with the URL of your Swagger file.

1. Log in to your portal as a user with admin or content creation privileges.
2. Select **Content > SmartDocs** in the Drupal administration menu.
3. Select the model you want to update.
4. Under **Operations**, select **Import**.
5. Select **Swagger JSON**or**Swagger YAML** in the **Choose Format**dropdown on the SmartDocs import page.
6. Select **File** or **URL** in the **Upload Type** dropdown (you must select **URL** for Swagger 1.2).
   1. If you select **File**, browse to the Swagger file.
   2. If you select **URL**, specify the URL of the Swagger file.
7. Select the **Import** button to import it into the model.
8. You are redirected to the model’s information page where you can now render the model.