

Practical: AWS API Gateway

Table of Contents

- [Creating a REST API](#)
- [Using in applications](#)
- [Delete unused resources](#)

API Gateway is a service abstraction that lets you create a unified API from multiple internal services. There are several types of API that you can create. For this unit we'll only be creating a REST API.

API gateways have multiple uses:

- Create a unified interface for multiple internal services
- Integrate with authentication (eg. Cognito) without having to implement it in your services
- Create a stable endpoint that you can point a DNS record to
- Manage multiple versions of an API

The main functionality of REST APIs in API Gateway use *resources*, which map particular paths and methods to services. For example, a resource might map **GET** request to **/files** to an internal service which responds with a list of a user's files. Once resources are defined, they are deployed to a *stage* which is a fixed version of the resources. You can have multiple stages available at the same time, each of which responds according to the resources that were deployed to it.

Creating a REST API

We'll create a simple REST API to demonstrate the basics of creating resources and stages. Our goal will be to pass a request to **/sun** on our API to an external API that responds with sunrise and sunset information for QUT on the current date. In most cases you will not map directly to an external API; this is just done here for demonstration without the need to create an internal service.

- Find *API Gateway* on the AWS console
- Click *Create an API*
- In the *REST API* bubble, click *Build*
- Add a name, such as *n1234657-practical*
- Click *Create API*

Your API is now created, but no resources have been configured. You should be directed to the *Resources* page for your API.

- Click *Create resource*
- Under *Resource name* enter **sun**

- Click *Create resource*
- You will now see the details for the `/sun` resource
- Click *Create method*
- Under *Method type* select *GET*. This is the HTTP method used when querying this resource. You can define multiple methods for the same resource.
- Under *Integration type* choose *HTTP*
- Under *HTTP method* choose *GET*. This is the HTTP method used when calling the endpoint.
- Under *Endpoint URL type* `https://api.sunrisesunset.io/json?lat=-27.477990&lng=153.027701&timezone=10&date=today`. This is an external API that provides sunrise and sunset information. The parameters point to QUT Gardens Point campus.
- Click *Create method*

Your resource has been defined, but not yet deployed so it is not accessible. We need to create a new stage and deploy to it.

- Click *Deploy API* in the top right
- Under *Stage* choose *New Stage*
- Under *Stagename* type `v1`
- Click *Deploy*


You should now see the details page for your stage.

- On the left, click the + symbol beside `v1`. This will expand the resources tree.
- Click on `/sun` to show the methods for this resource
- Click on *GET*.
- Copy the *Invoke URL* by clicking the blue box beside the URL
- Paste into the URL bar in a new tab on your browser and press Enter

You should see the response from the external URL. Note that the URL bar still displays the URL from the API gateway. The response was processed through the API gateway, rather than redirecting the browser.

Using in applications

API Gateway is a service abstraction that allows you to have a single endpoint for several services. Some uses:

- Present a single endpoint to clients if you have several microservices that clients need to access
- Access Lambda functions (coming later!) via HTTP requests
- Add authentication to endpoints (configure in the *Authorization* tab.) See [Integrate API Gateway with Cognito](https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-integrate-with-cognito.html)  (<https://docs.aws.amazon.com/apigateway/latest/developerguide/apigateway-integrate-with-cognito.html>) to see how this can be done with Cognito.

- Add HTTPS access to your endpoints (requires coordination with Route53 and the certificate manager, coming later!)
- Rate limiting
- Create endpoints for sending messages with SQS and other services

Please note that currently students don't have permission to add tags, so API gateways are exempt from our usual policy that all resources should be tagged with a `purpose` tag. The `qut-username` tag is added automatically, but unfortunately there is no way for students to add the `qut-username2` tag to allow a partner. We will update this documentation if this changes. If this is a problem for project collaboration, please email `cab432@qut.edu.au` to request your partner's username to be added.

Delete unused resources

If you are done with your API gateway, please delete it.

TEQSA PRV12079 | CRICOS 00213J | ABN 83 791 724 622