# Publish your own API

AppSynergy allows you to publish a custom API where you can respond to RETful API calls using PL/SQL. This works by firing a trigger on the *WebRequest* table in your database each time an API call is received. Your trigger simply reads the request and generates a response. You can also use this feature to process inbound Webhooks.

## Example

Create a BEFORE UPDATE trigger on the *WebRequest* table via *Tools > Database Triggers*. The basic code for your trigger should look something like this:

```
BEGIN

  -- only respond if resp_code is NULL
  IF (NEW.resp_code IS NULL) THEN

    -- verify the request is from a valid API user (optional)
    IF (NEW.req_user_id != 'u1234') THEN
      SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Permission Denied.';
    END IF;

    -- parse req_params and/or req_body to generate your response
    IF (JSON_VALUE(NEW.req_params,'$.TestMe') = 'true') THEN

      SET NEW.resp_code = 200;
      SET NEW.resp_body = JSON_OBJECT('MyResponse','Hello World');

    ELSE
      SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Invalid Request. Try this: TestMe=true';
    END IF;

  END IF;

END
```

Your trigger will fire for every API call received. Your code should first test that *NEW.resp_code* is NULL, then set *NEW.resp_code* and *NEW.resp_body* to appropriate values for the response. Typically this is all your code needs to worry about.

If an error is encountered when the trigger fires it will be rolled back and the system will then update the row a second time with a non-null *resp_code* and *resp_body* to log the error. The best way for your code to throw an error is via the SIGNAL command (as shown above).

You can query the *WebRequest* table for diagnostic data.

## API Specification

**Request**

```
GET | POST https://www.appsynergy.com/api?action=WEB_REQUEST&apiKey=YOUR_API_KEY&OptParam1=OptValue1

{
  "AnyValidJson": "OK"
}
```

Your trigger will see the request data as follows:

- The *NEW.req_user_id* field contains the user id associated with the API Key that made the request.
- The *NEW.req_method* field contains the request method; only GET and POST are supported.
- The *NEW.req_params* field contains any optional URL parameters represented as a single JSON object with name:value pairs.
- The *NEW.req_headers* field contains the request headers as a JSON object.
- The *NEW.req_body* field contains the request body; this must be a JSON object if specified.

**Response**

```
{
  "AnyValidJson": "OK"
}
```

Your trigger should SET the response fields as follows:

- Set the *NEW.resp_code* field to an appropriate value; use *200* for success and *400* for error. If your code throws a SIGNAL at any level AppSynergy will catch that and respond with a 400 response code and use the message text of the SIGNAL statement as the error message.
- Set the *NEW.resp_body* field to an appropriate value; the response can be any valid JSON object.
- Optionally set the *NEW.resp_headers* field to a valid JSON object containing any additional response headers you may wish to include. The default response header is *{"content-type":"application/json; charset=utf-8"}*. To append additional response headers use: *SET NEW.resp_headers = JSON_MERGE(NEW.resp_headers, JSON_OBJECT('x-parasql-test','testvalue'));*