Project Tortoise

REST API Documentation

INTRODUCTION

Building a back-end API layer introduces a whole new layer of coordination between server and client code. While there are many aspects to this delicate dance of communication, one key ingredient to minimizing back-and-forth-confusion-about what-call-does-what, is consistently communicating about your API endpoints.

This is by no means a rocket science, but over time we’ve created a template that we now tend to use and have been out to share. Conveniently when the time comes to publish an API externally, this serves as an invaluable tool for creating public documentation.

|  |  |
| --- | --- |
| **Title** | ***Example : Show All Users***   * *Note: try to use verbs that match both request type (fetching vs modifying) and plurality (one vs multiple.)* * *Note: Also add additional info here such as a description, if need be.* |
| **URL** | *<The URL structure (path only, no root url)>*  **/users or /users/:id or /users?id=:id**   1. *For fixed urls: /users or /photos* 2. *For urls with parameters in them: /users/:id or /photos/:photo\_id or /users?id=:id* |
| **Method** | ***GET | POST | DELETE | PUT*** |
| **URL Params** | *<If URL params exist, specify them in accordance with name mentioned in URL section. Separate into optional and required.>*   **Required:**  id=[integer]  example: id=12   **Optional**  photo\_id=[alphanumeric]  example: photo\_id=2345kj3 |
| **Data Params** | *<If making a post request, what should the body payload look like? This is a good time to document your various data constraints too.>*  **Example:**  {  u : {  email : [string],  name : [string],  current\_password : [alphanumeric]  password : [alphanumeric],  password\_confirmation : [alphanumeric]  }  }  **Example:**  {  u : {  email : "john@smith.com",  name : "John",  current\_password : "apassw0rd"  password : "anewpassw0rd",  password\_confirmation : "anewpassw0rd"  }  } |
| **Success Response** | *<What should the status code be on success and is there any returned data? This is useful when people need to to know what their callbacks should expect!>*  **Example:** **Code:** 200  **Content:** { id : 12 } |
| **Error Response** | *<Most endpoints will have many ways they can fail. From unauthorized access, to wrongful parameters etc. All of those should be listed here. It might seem repetitive, but it helps prevent assumptions from being made where they shouldn't be.>*  **Example:** **Code:** 401 UNAUTHORIZED  **Content:** { error : "Log in" }  OR **Code:** 422 Unprocessable Entry  **Content:** { error : "Email invalid" } |
| **Sample Call** | *<Just a sample call to your endpoint in a runnable format ($.ajax call or a curl request) - this makes life easier and more predictable.>*  $.ajax({  url: "/users",  dataType: "json",  data : {  u: {  id : 12,  email : "john@smith.com"  }  },  type : "PUT",  success : function(r) {  console.log(r);  }  }); |
| **Notes** | *<This is where all uncertainties, commentary, discussion etc. can go. I recommend timestamping and identifying oneself when leaving comments here.>* |