

Use Map and JSON.serialize to mock HTTP Response

Did you ever try to mock an HTTP response?

Perhaps you did something like this:

```
HttpResponse response = new HttpResponse();
response.setHeader('Content-Type', 'application/json');
response.setBody('{ "name": "my_username", "first-name": "My", "email": "user" + UserInfo.getUserId() +
"@example.test", "attributes": { "rel": "edit" } }');
response.setStatusCode(200);
return response;
```



This is the most common approach, but...

You can make it cleaner with Map and JSON.serialize.



```
HttpResponse response = new HttpResponse();
response.setHeader('Content-Type', 'application/json');
response.setBody(JSON.serialize(
    new Map<String, Object>{
        'name' => 'my_username',
        'first-name' => 'My',
        'email' => 'user' + UserInfo.getUserId() + '@example.test',
        'attributes' => new Map<String, Object>{
            'rel' => 'edit'
        }
    }
));
response.setStatusCode(200);
return response;
```



Since the HTTP Body is a JSON, in this way, the code is as much similar/formatted as standard JSON as possible.

There is no need to create wrappers.
Just a simple Map, JSON.serialize, and voilà!