CREATING FAVORITE QUESTION & ACCEPT ANSWER ENDPOINTS

In this lesson we're going to create api endpoints for favoriting question and accepting an answer as best anser. Since we've already had that logics in our existing controller, so we're gonna copy the routes as well as the controllers, then make a bit changes to them and then make some tests in postman.

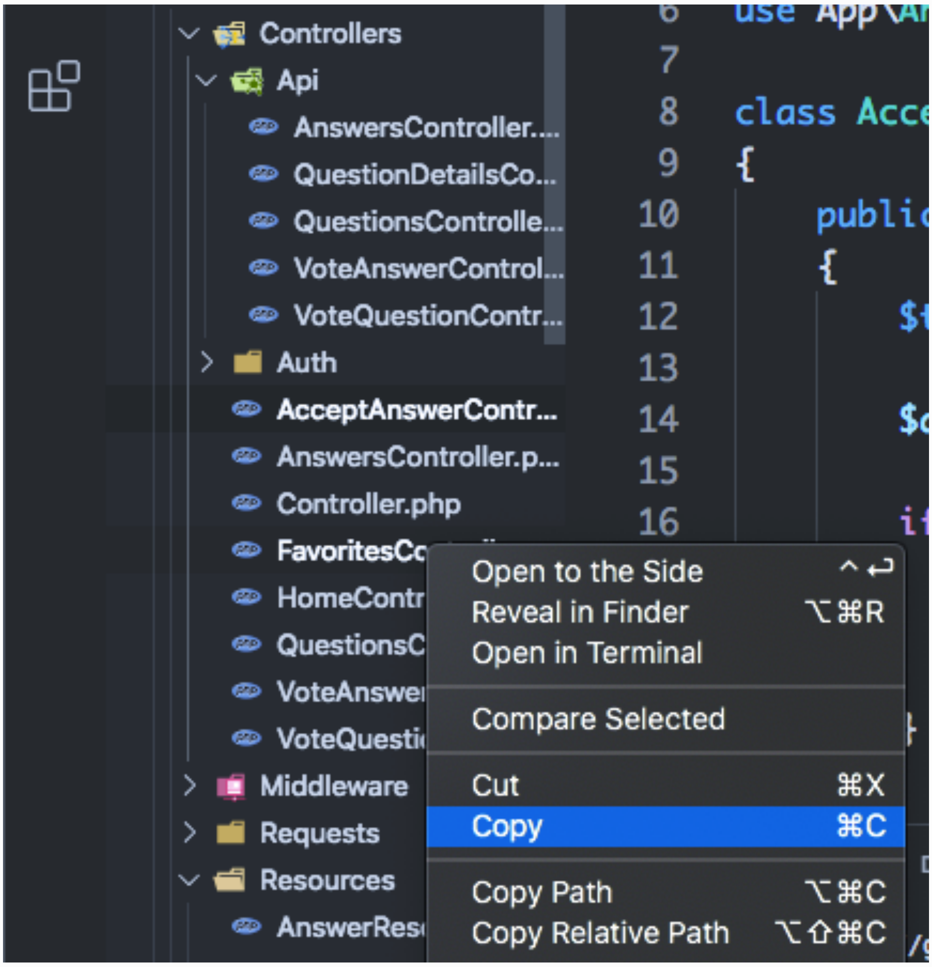
Before we beggin let's go ahead and open up our terminal and create a new branch:

```
git checkout -b lesson-53
```

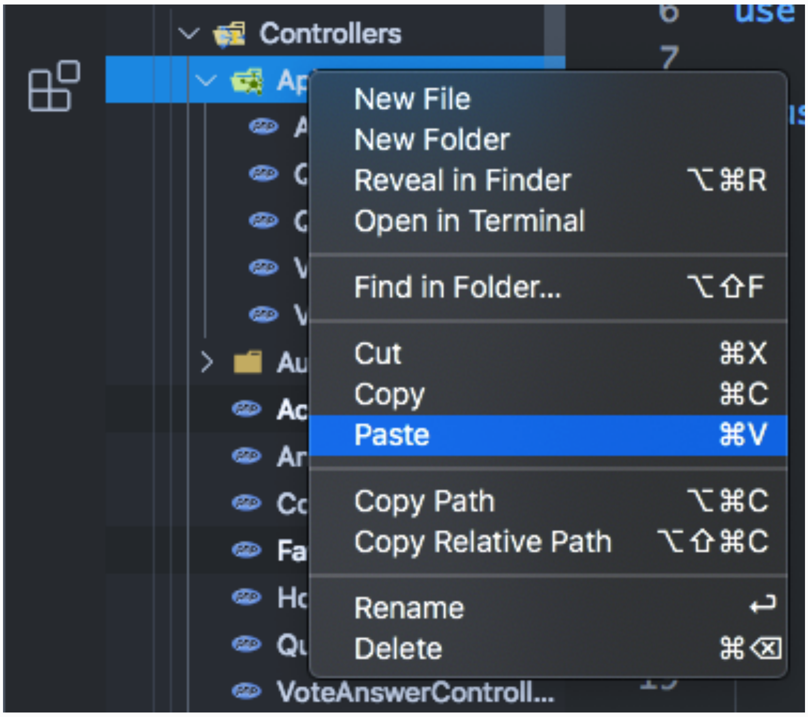
CREATING THE ENDPOINTS

1. Copy-Paste the FavoriteController & AcceptAnswerController

Alright, let's navigate to `app/Http/Controller` directory. Inside this directory let's copy the `FavoriteController` and `AcceptAnswerController` files.



Go to `Api` subdirectory then paste them inside it.



Firstly, let's open the `FavoriteController` then fix the namespace stuffs. Then let's get rid of the constructor. And remove all if statements as well as the `return back()` statement on all methods.

```
# Api/FavoriteController.php
<?php

namespace App\Http\Controllers\Api;
use Illuminate\Http\Request;
use App\Question;
use App\Http\Controllers\Controller;

class FavoritesController extends Controller
{
    public function store(Question $question)
    {
        $question->favorites()->attach(auth()->id());

        return response()->json(null, 204);
    }

    public function destroy(Question $question)
    {
        $question->favorites()->detach(auth()->id());

        return response()->json(null, 204);
    }
}
```

Alright, let's do almost the same thing to `AcceptAnswerController`. At the top let's add `api` in the controller namespace. Then import the base `Controller` namespace. And then remove the if statement as well as the return back.

```
# Api/AcceptAnswerController.php
<?php

namespace App\Http\Controllers\Api;

use Illuminate\Http\Request;
use App\Http\Controllers\Controller;
use App\Answer;

class AcceptAnswerController extends Controller
{
    public function __invoke(Answer $answer)
    {
        $this->authorize('accept', $answer);

        $answer->question->acceptBestAnswer($answer);

        return response()->json([
            'message' => "You have accepted this answer as best answer"
        ]);
    }
}
```

2. copy rute dari `webp.php` ke `api.php`

Alright, Let's open up the `web.php` file. Then let's copy these three routes.

```
Route::post('/answers/{answer}/accept', 'AcceptAnswerController')->name('answers.accept');
Route::post('/questions/{question}/favorites', 'FavoritesController@store')->name('questions.favorite');
Route::delete('/questions/{question}/favorites', 'FavoritesController@destroy')->name('questions.unfavorite');
```

Let's open up the `api.php` file, then paste them inside the `auth:api` middleware group. Adjust the controller path to `Api`. Then remove all `name` from our routes since we're no longer need that for our api.

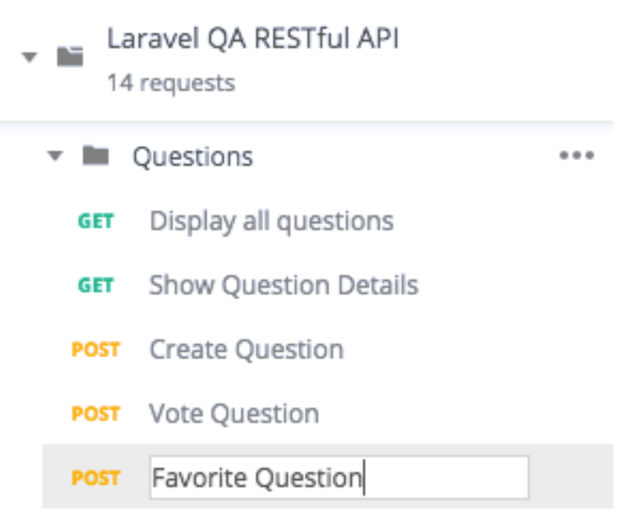
```
# routes/api.php
Route::middleware(['auth:api'])->group(function() {
    // ...
    Route::post('/answers/{answer}/accept', 'Api\AcceptAnswerController');
    Route::post('/questions/{question}/favorites', 'Api\FavoritesController@store');
    Route::delete('/questions/{question}/favorites', 'Api\FavoritesController@destroy');
});
```

Alright, let's save all changes. And now let's thest these endpoints out in Postman.

TESTING THE ENDPOINTS IN POSTMAN

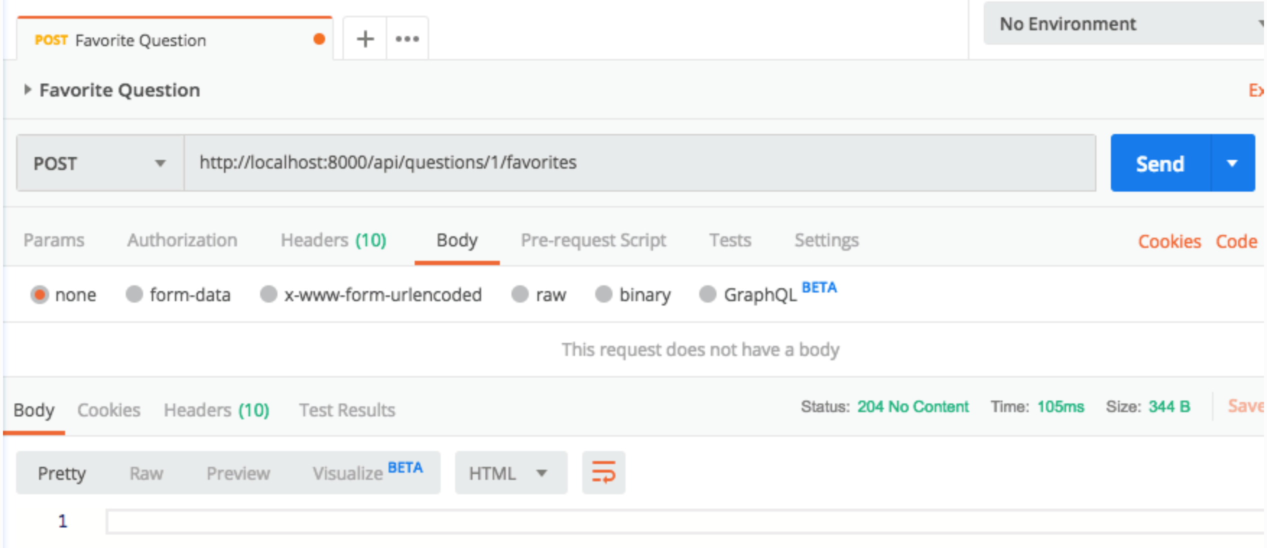
1. Testing the Favorite Question endpoint

Let's head over to postman. Then go to `Questions` folder. Open the `Vote question` request, then save it as `Favorite question`.



The method is gonna be `post` and the url is gonna be `~/api/questions/1/favorites`. You can change the question id to any existing number that you have. In my case I'll set it to `1`.

The headers will be the same, so you don't need to touch anything in there. You can optionally choose the **none** option in the **Body** section.



POST http://localhost:8000/api/questions/1/favorites

● none

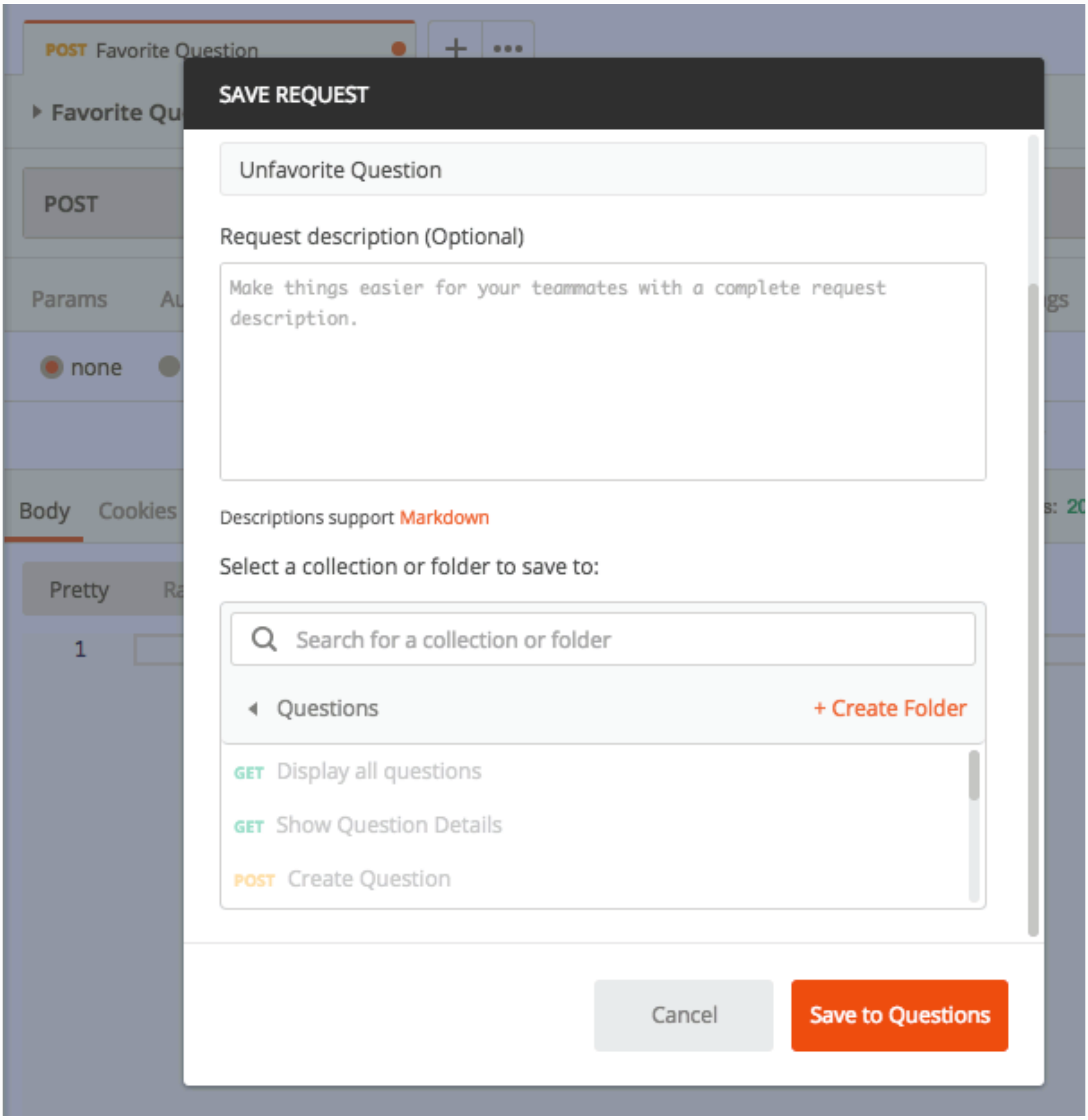
This request does not have a body

Status: 204 No Content Time: 105ms Size: 344 B

If you hit the **Send** button. Make sure you're getting back **204 No Content** response wich is the expected response.

2. Testing the Unfavorite Question endpoint

Now let's make another request to test unfavorite question. Let's go ahead and hit the **save** button on the top right. Choose **Save as** option and save it as `Unfavorite question`. Choose the `Question` folder then hit **Save to Questions** button to save the request to `Questions` folder.



POST Favorite Question
Favorite Qu
POST
Params
Au
none
Body
Cookies
Pretty
Ra
1
SAVE REQUEST

Unfavorite Question

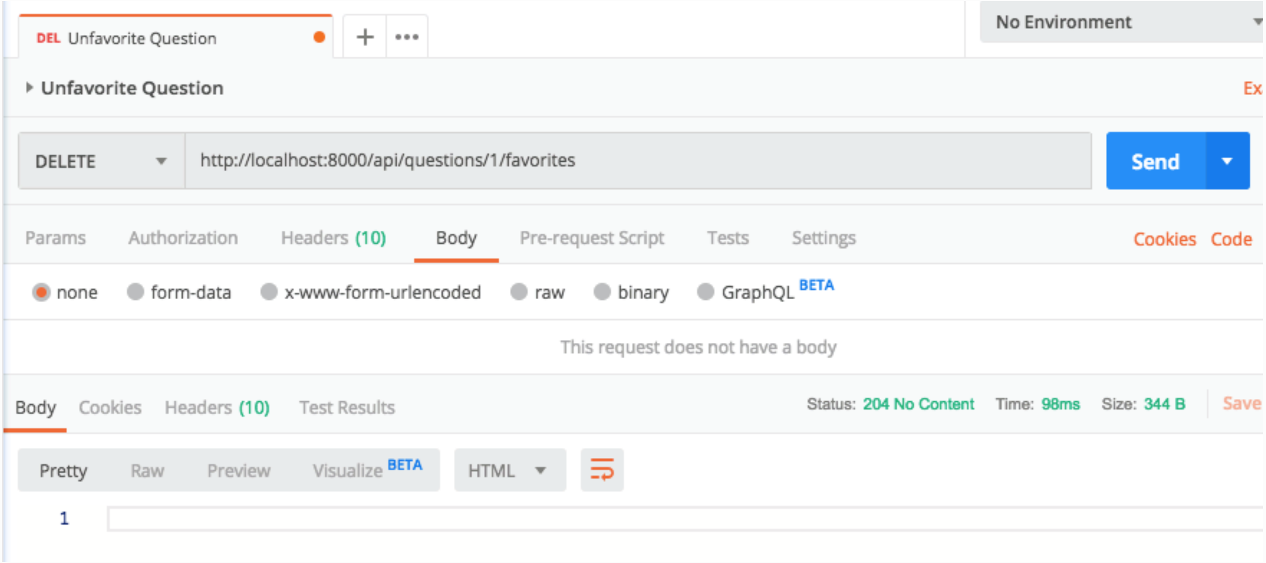
Request description (Optional)
Make things easier for your teammates with a complete request description.
Descriptions support Markdown
Select a collection or folder to save to:
Search for a collection or folder
Questions
+ Create Folder

GET Display all questions

GET Show Question Details

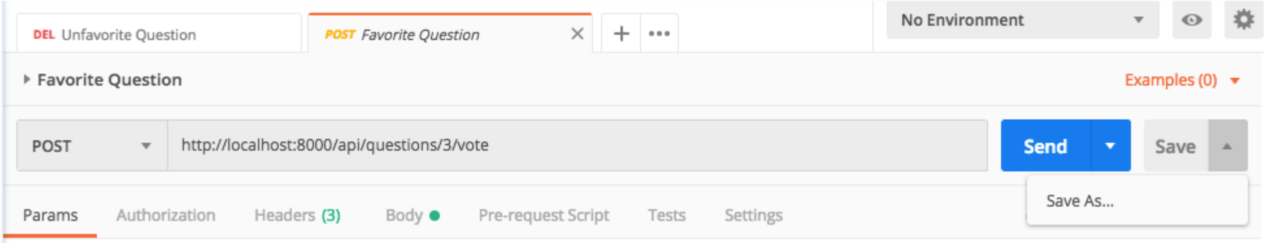
POST Create Question

Let's change the HTTP method from `post` to `delete`. While the url and other things still the same. When you hit the **Send** button. You're getting back **204 No Content** response, awesome.

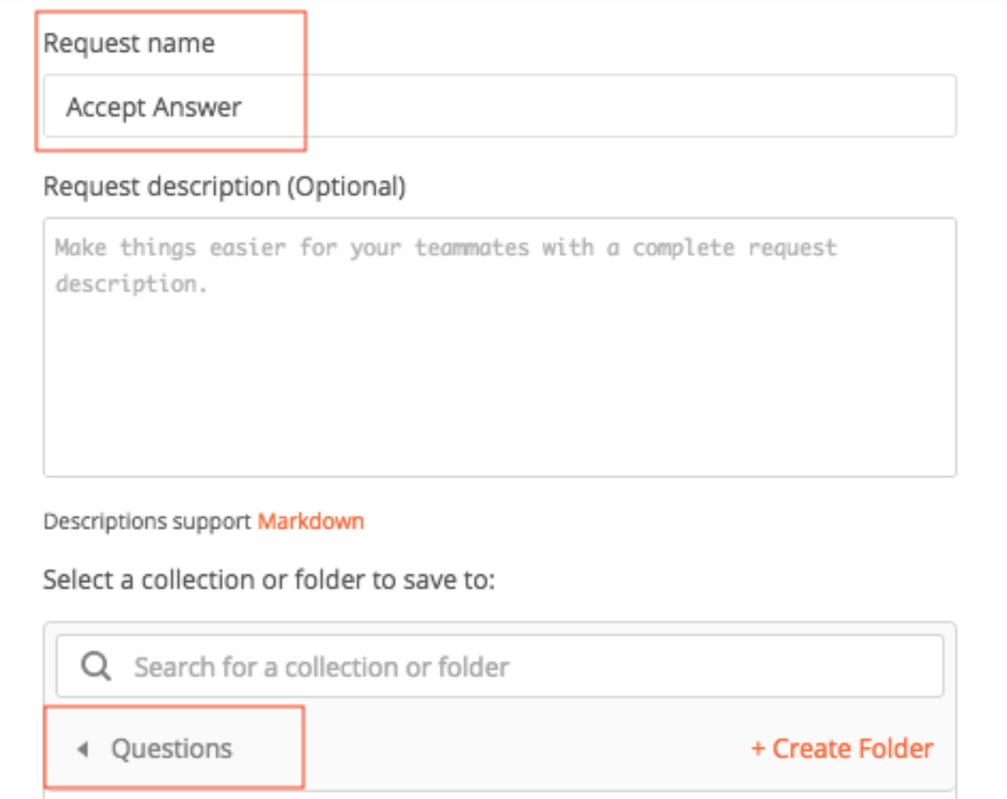

DEL Unfavorite Question
No Environment
Unfavorite Question
DELETE
http://localhost:8000/api/questions/1/favorites
Send
Params Authorization Headers (10) Body Pre-request Script Tests Settings
Cookies Code
● none ○ form-data ○ x-www-form-urlencoded ○ raw ○ binary ○ GraphQL BETA
This request does not have a body
Body Cookies Headers (10) Test Results
Status: 204 No Content Time: 98ms Size: 344 B
Save
Pretty Raw Preview Visualize BETA HTML

3. Testing the Accept Answer endpoint

Last, let's make another test for testing our accept answer endpoint. Let's reopen the `Favorite Question` request. Then go to the top right and hit the **Save** button and then choose **Save as** button.


DEL Unfavorite Question
POST Favorite Question
No Environment
Favorite Question
Examples (0)
POST
http://localhost:8000/api/questions/3/vote
Send
Save
Save As...
Params Authorization Headers (3) Body Pre-request Script Tests Settings

Change the **Request Name** to `Accept answer`. Hit the `Questions` in the top of **Select a collection** section to go to the root.



Request name

Accept Answer

Request description (Optional)

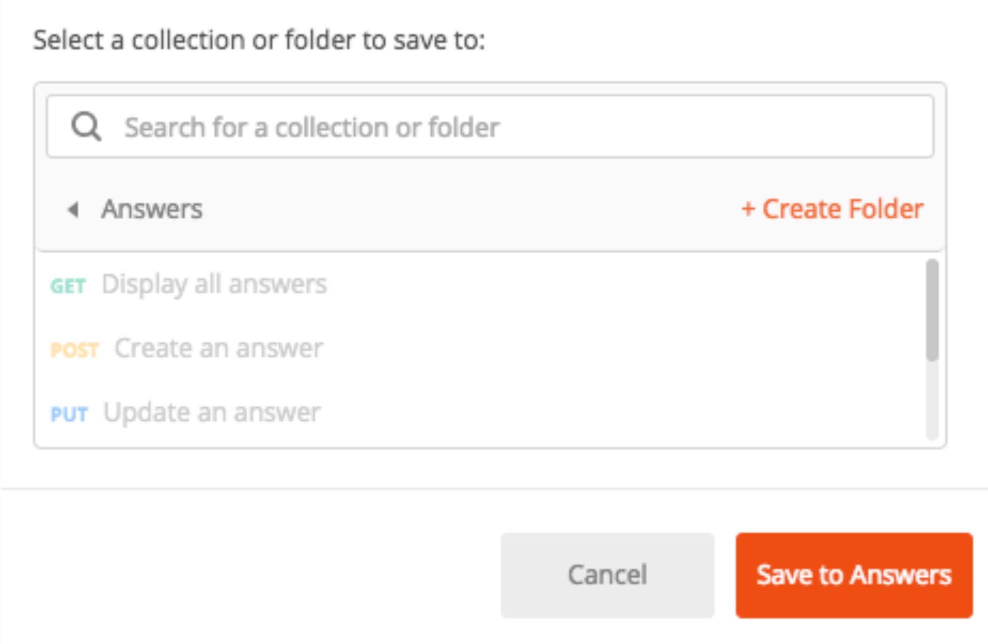
Make things easier for your teammates with a complete request description.

Descriptions support Markdown

Select a collection or folder to save to:

Search for a collection or folder

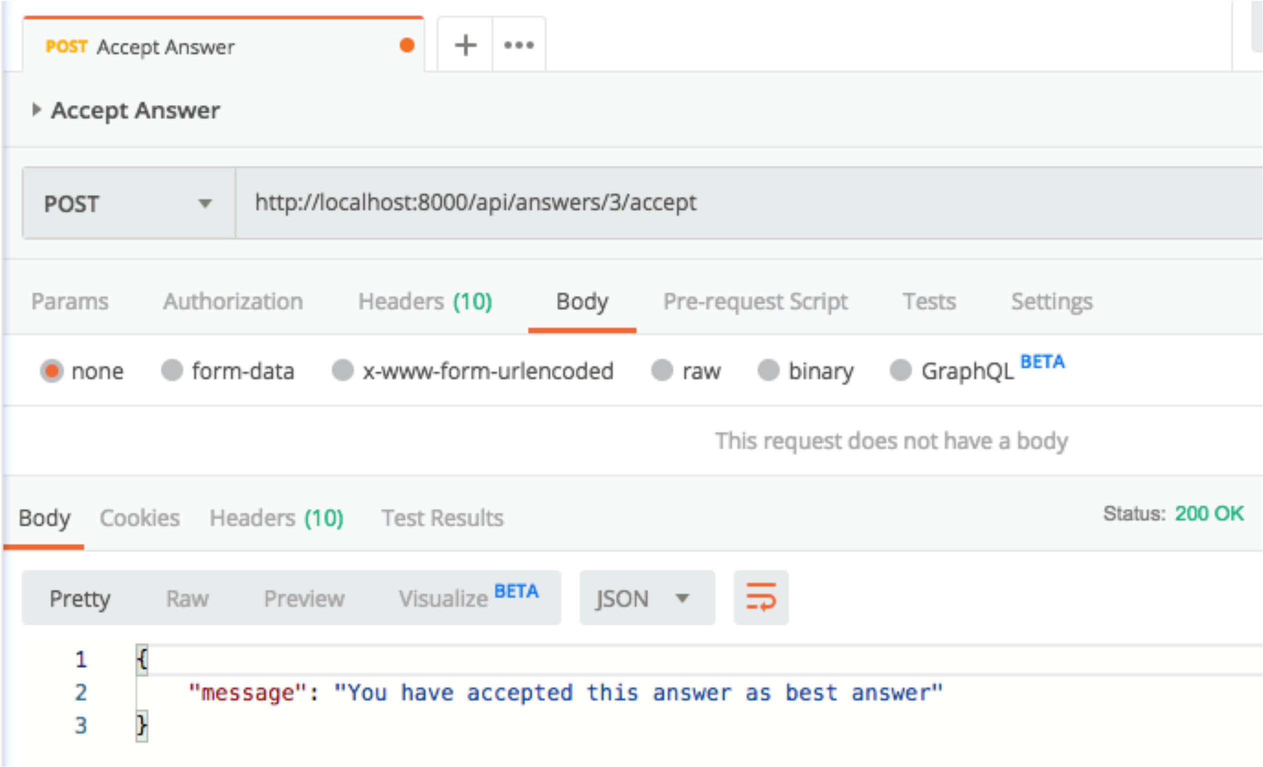
Then select the `Answers` folder to put the test request in Answers folder. Then hit the **Save to Answers** button.



Select a collection or folder to save to:

Search for a collection or folder

The last thing that you need to do is to change the url to `~/api/answers/1/accept`. When you hit the **Save** button you're getting back the success message.



POST http://localhost:8000/api/answers/3/accept

This request does not have a body

Status: 200 OK

```
{
    "message": "You have accepted this answer as best answer"
}
```

SUMMARY

In this lesson, we looked at how to create an api endpoint to mark a certain question as farorite question and and api endpoint for accepting answer. In the next one we're going to create additional endpoint which is going to use to retrieve current user's posts. So this endpoint kind of summary of what questions and answers that current user's logged in have been created.

```
git add .
git commit -m "Create api endpoints for favoriting the question &
accepting answer"
git push origin lesson-53
```