**Vastra API – Endpoints**

|  |  |
| --- | --- |
| **Created Date** | **28-08-2023** |
| **Created By** | **Sumit Ranjan** |

**Pre-Requisites to test code and API:**

* Visual Studio 22
* MS SQL Server Express 16 or higher
* Postman

**Steps to clone repository and run project:**

1. Clone the repo to your local machine using command:

*git clone https://github.com/vastra-dev01/Vastra.git*

1. Create ‘appsettings.Development.json’ file in project directory with ‘appsettings.json’ and copy paste the content of ‘appsettings.json’ to it. Then fill the values which are empty in ‘ConnectionStrings’ and ‘SampleUsers’ sections.
2. Open ‘.sln’ file in Visual Studio 22. Open package manager console and run following commands

add-migration {migration-name}

update-database

1. Run the project in debug mode in VS 22.
2. It will open swagger home page in browser where all endpoints are visible.

**Endpoints open for all:**

Authentication:

* [Post] api/authentication/authenticate

Categories:

* [Get] api/categories
* [Get] api/categories/{categoryId}
* [Get] api/categories/{categoryId}/subCategories

Products:

* [Get] api/categories/{categoryId}/products
* [Get] api/categories/{categoryId}/products/{productId}

Users:

* [Post] api/roles/{roleId}/users (duplicate user check will be performed in addition to validation)

**Note :** Above end-points are accessible to anyone without authN/authZ (validation will be performed on data which is sent in req body) and all other endpoints except these are protected by authN/authZ rules in addition to validation.

**Steps for accessing the protected endpoints:**

1. On the swagger homepage , expand the ‘[Post] api/authentication/authenticate’ endpoint.
2. Select ‘Try it out’.
3. Set request body data type to ‘application/json’ in drop-down.
4. In responses section, set ‘Media type’ to ‘application/json’ in drop-down.
5. Fill request body with ‘phoneNumber’ and ‘password’.(Provide correct credentials for user/admin as you had filled in ‘appsettings.Development.json’).
6. Click ‘Execute’.
7. You will get a json web token with 200 Ok response.
8. Copy and save the token.
9. Now, whenever you send request to any protected endpoint, send the token as bearer token in authorization header of the request.

**Authorization/Authentication Rules:**

1. There are two types of roles: Admin & User.
2. Any user’s role will be either ‘Admin’(roleId : 1) or ‘User’(roleId : 2). From the endpoints which are protected , some need Admin credentials and others need User credentials to be accessed.

**Admin AUTHZ Endpoints:**

Categories:

* [Post] /api/categories
* [Put] [Patch] [Delete] /api/categories/{categoryId}

Products:

* [Post] /api/categories/{categoryId}/products
* [Put] [Patch] [Delete] /api/categories/{categoryId}/products/{productId}

Roles

* All endpoints.

Users

* [Get] /api/roles/{roleId}/users

**Note:** All other **protected endpoints** need User AuthN & AuthZ. Also, currently admin may be able to access the endpoints with User AuthZ like addresses or orders endpoints, which means admin will be able to create address or orders. This can be blocked in future as per requirement.

Also, for endpoints with ‘User’ authZ , the phone number of the user whose jwt is being sent as bearer token should match the user whose userId is being used in request param i.e. any user will be able to access resources related to him/her only, not of other users.

**Validation for data in request body of POST requests:**

* **Addresses** : [Post] */api/roles/{roleId}/users/{userId}/addresses/{addressId}*

{

"**location**": required | length 10-100 | alphanumeric values, - , and space allowed.

"**city**": required | length 3-50 | should be a single word (no numbers or special characters).

"**state**": required |length 3-50|should be a single word (no numbers or special characters).

"**pinCode**": required | length 6| a valid pin code.

"**country**": required|length 3-50|should be a single word (no numbers or special characters).

"**tag**": length 3-50 | alphabets, numbers, -, and \_ allowed.

"**type**": Type can only be 0(Primary) or 1(Secondary)

}

**Note :** Same applies for any of these values present in [PUT] or [PATCH] requests.

* **CartItems:** [Post] */api/roles/{roleId}/users/{userId}/orders/{orderId}/cartItems/{cartItemId}*

{

"**productId**": {int} required | length 1-10 | a valid product id

"**quantity**": {int} required | length 1-10

}

**Note :** Same applies for any of these values present in [PUT] or [PATCH] requests.

* **Categories :** [Post] */api/categories*

{

"**categoryName**": {string} required | max length 50 | alphabets separated by – or space

}

**Note :** Same applies for any of these values present in [PUT] or [PATCH] requests.

* **Products:** [Post] */api/categories/{categoryId}/products*

{

"**name**": {string} required | length 8-50 | alphanumeric values separated by - or space"

"**description**": {string} | length 20-200 | alphanumeric values separated by - or space"

"**size**": required | Only M|S|L|XS|XL|XXL|XXXL allowed

"**colour**": {string} | length 2-50 | alphabets separated by - or space

"**sku**": {string} required | a valid 8 digit sku with first 4 characters and next 4 numbers like – ‘PUPU1010’

"**price**": {float} required | a decimal or normal number like 234.90 or 234 both will work.

"**image**": {string} | maxlength 100 | a valid filepath ending with a image extension like “/test/test/img.jpeg” (png | jpg | jpeg allowed)

"**quantity**": {int} required | length 1-10

}

**Note :** Same applies for any of these values present in [PUT] or [PATCH] requests.

* **Roles :** [Post] */api/roles*

{

"**roleName**": {string} required | length 3-50 | only alphabets, space or - .

}

**Note :** Same applies for any of these values present in [PUT] or [PATCH] requests.

* **Users:** [Post] */api/roles/{roleId}/users*

{

"**firstName**": {string} required | length 3-50 | a single word only (no numbers or special chars)

"**lastName**": {string} length 3-50 | a single word only (no numbers or special chars)

"**phoneNumber**": {string} a valid 10 digit phone number

"**password**": {string} required | should be between 5-15 characters containing at least one special character(!,#,@ or %)

"**emailId**": {string} | a valid string which looks like an email id – test@test.com

}

**Note :** Same applies for any of these values present in [PUT] or [PATCH] requests.