CIS 424

Lecture 8: Server-Side

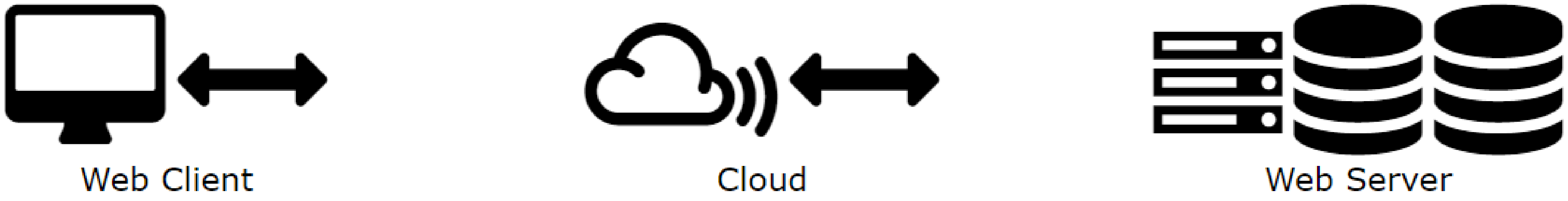
Mobile Application and Development

Prof. Dr. Mohamed Ezz

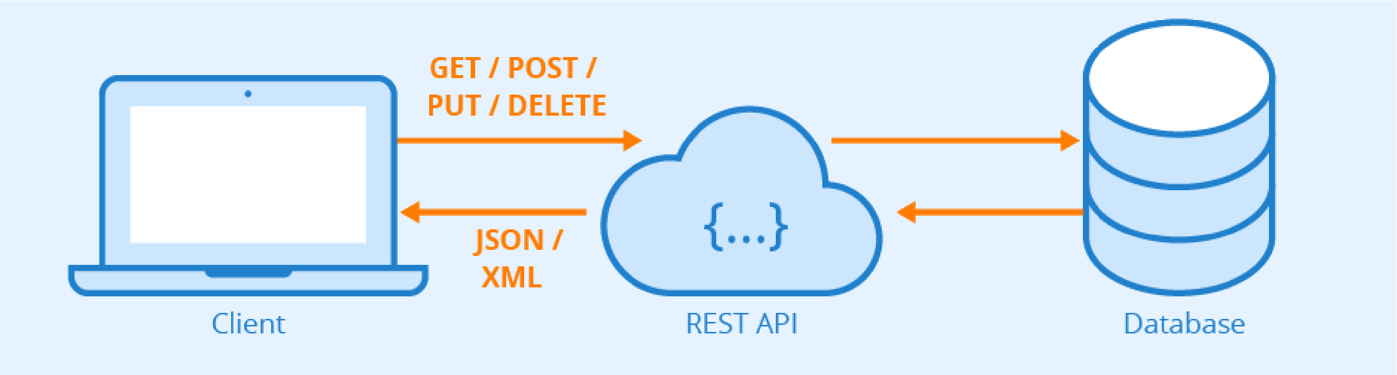
Server-Side Technologies

Concepts

- **HTTP** stands for **H**yper **T**ext **T**ransfer **P**rotocol
  - Communication between client computers and web servers is done by sending **HTTP Requests** and receiving **HTTP Responses**
- **REST** : stands for **RE**presentational **S**tate **T**ransfer.
  - REST is web standards based architecture and uses **HTTP** Protocol.
  - It revolves around resource where every component is a resource and a resource is accessed by a common interface using HTTP standard methods.



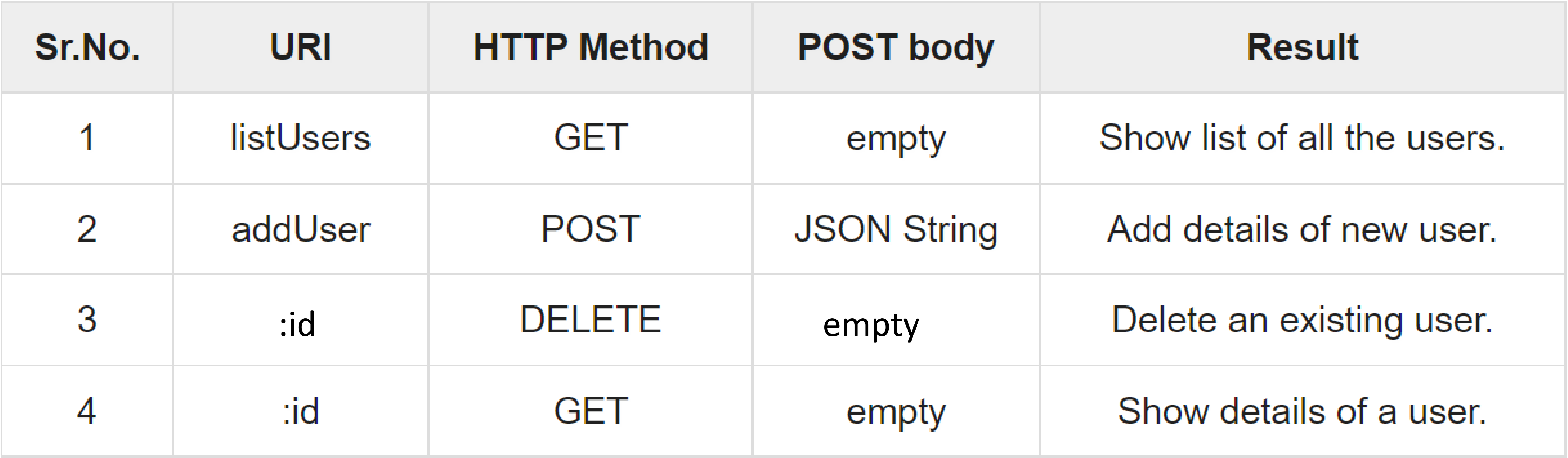
REST architecture



- Following four HTTP methods are commonly used in REST based architecture.
  - **GET** – This is used to provide a read only access to a resource.
  - **POST** – This is used to create a new resource.
  - **PUT** – This is used to update a existing resource or create a new resource.
  - **DELETE** – This is used to remove a resource.

- A **web service (WS)** is a collection of open protocols and standards used for exchanging data between applications or systems.
  - The WS is interoperability (e.g., communication between Java and Python, or Windows and Linux applications) is due to the use of open standards.
  - Web services based on REST Architecture are known as **RESTful web services**.
  - These webservices uses HTTP methods to implement the concept of REST architecture.
  - A RESTful web service usually defines a **URI, Uniform Resource Identifier** a service, which provides resource representation such as JSON and set of HTTP Methods.

We are going to provide following RESTful APIs

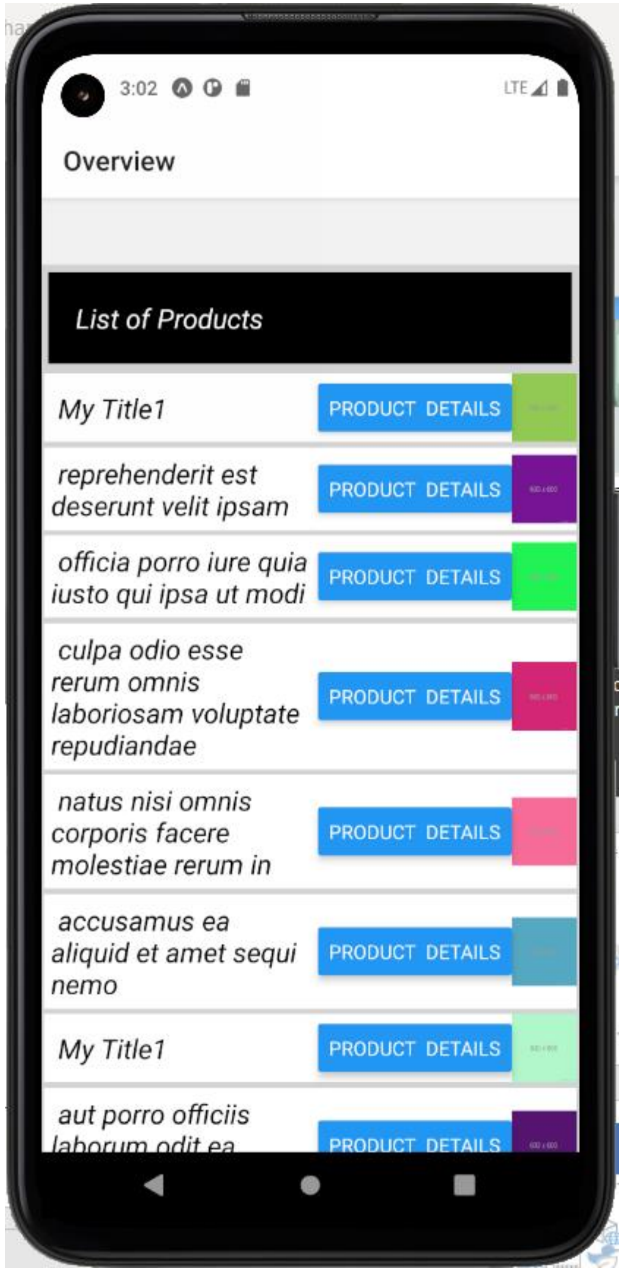


Sr.No.	URI	HTTP Method	POST body	Result
1	listUsers	GET	empty	Show list of all the users.
2	addUser	POST	JSON String	Add details of new user.
3	:id	DELETE	empty	Delete an existing user.
4	:id	GET	empty	Show details of a user.

React Native → RESTful APIs

Get All products

```
function ProductListScreen({ navigation }) {
  const [data, setData] = useState('');
  useEffect(() => { //when Did Mount
    var options = {
      method: 'GET',
      headers:
      { 'cache-control': 'no-cache',
        'x-apikey': 'e10a33c60bc1268c84e48cada806009efe740' } };
    fetch('https://cis424-ab14.restdb.io/rest/products', options)
      .then((response) => response.json())
      .then((responseJson) => {
        console.log(responseJson);
        console.log(Date());
        setData(responseJson);
      })
      .catch((error) => {
        console.error(error);
      }); }, []); //useeffect end brackets add [] to avoid
```



My Title1

reprehenderit est deserunt velit ipsam

officia porro iure quia iusto qui ipsa ut modi

culpa odio esse rerum omnis laboriosam voluptate repudiandae

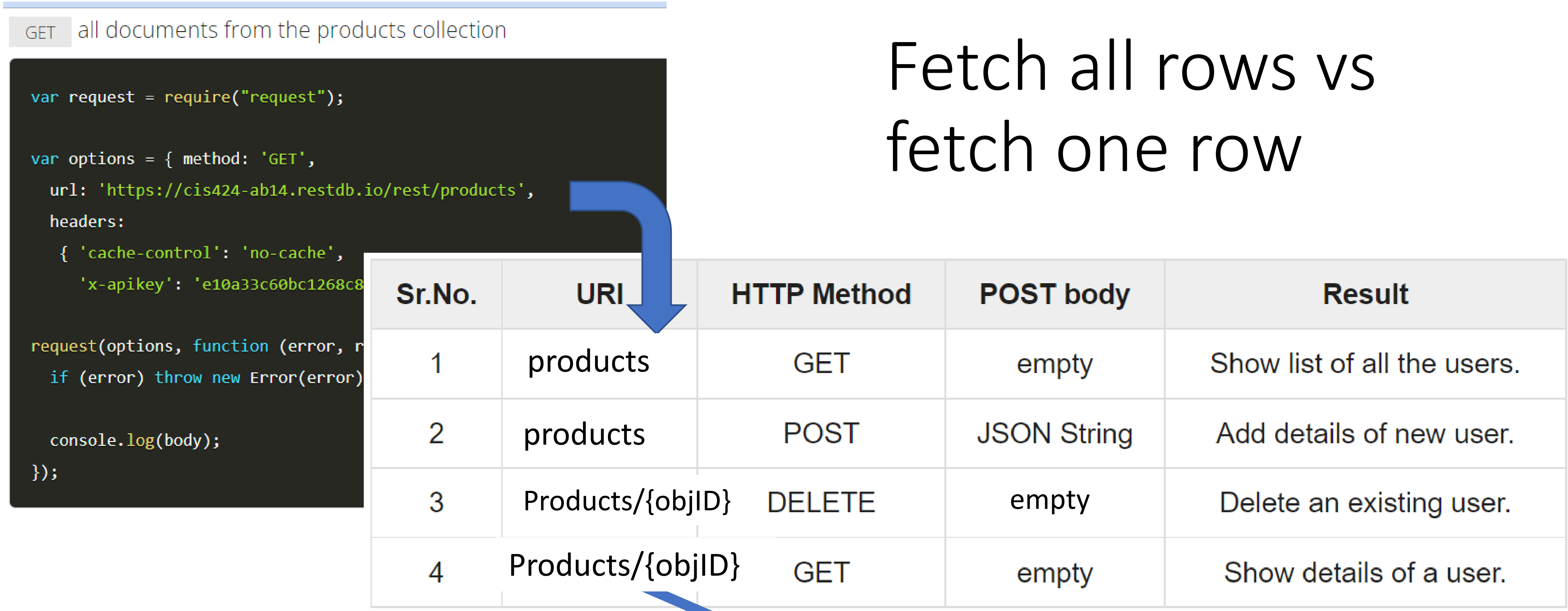
natus nisi omnis corporis facere molestiae rerum in

accusamus ea aliquid et amet sequi nemo

My Title1

aut porro officiis laborum odit ea

Fetch all rows vs fetch one row



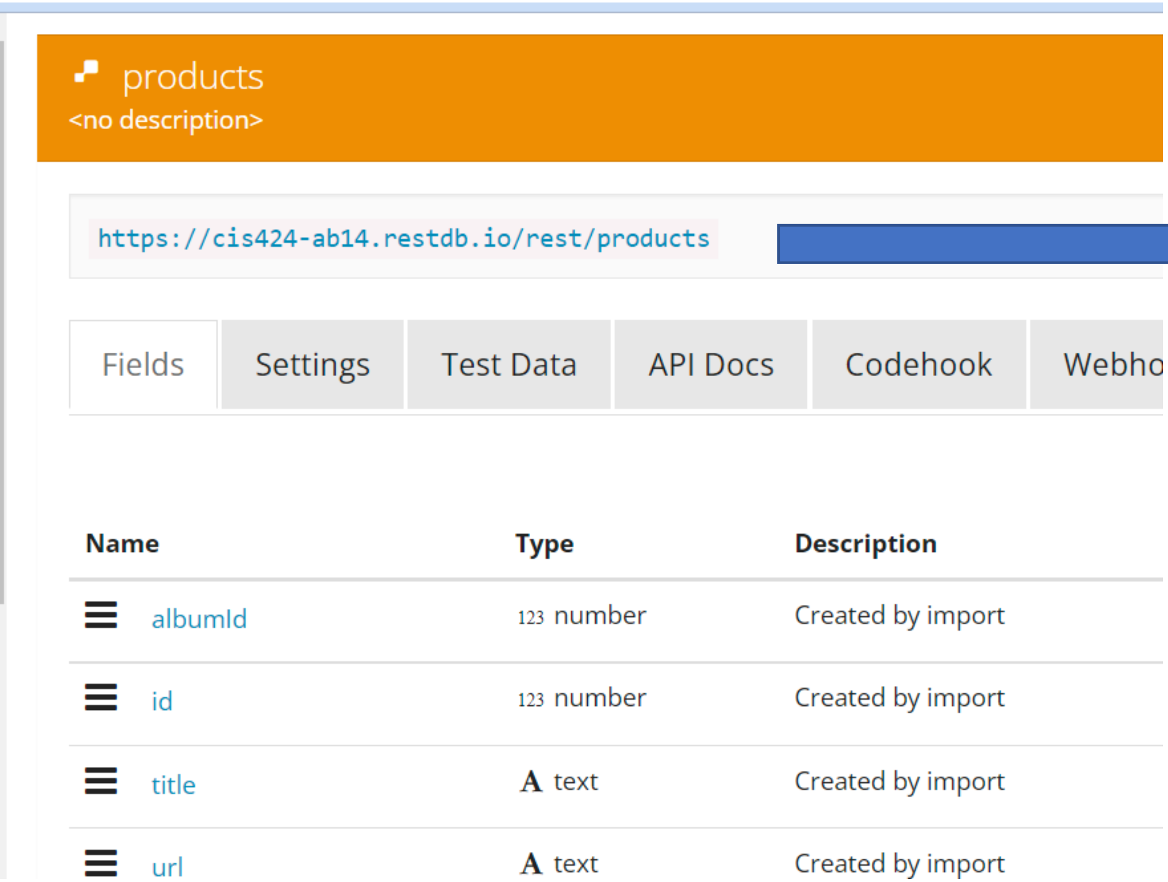
GET all documents from the products collection

```
var request = require("request");

var options = { method: 'GET',
  url: 'https://cis424-ab14.restdb.io/rest/products',
  headers:
   { 'cache-control': 'no-cache',
     'x-apikey': 'e10a33c60bc1268c8

request(options, function (error, r
  if (error) throw new Error(error)

  console.log(body);
});
```

Sr.No.	URI	HTTP Method	POST body	Result
1	products	GET	empty	Show list of all the users.
2	products	POST	JSON String	Add details of new user.
3	Products/{objID}	DELETE	empty	Delete an existing user.
4	Products/{objID}	GET	empty	Show details of a user.

https://cis424-ab14.restdb.io/rest/products/61744d8d868101090007b477

ObjectID of restdb



products

<no description>

`https://cis424-ab14.restdb.io/rest/products`

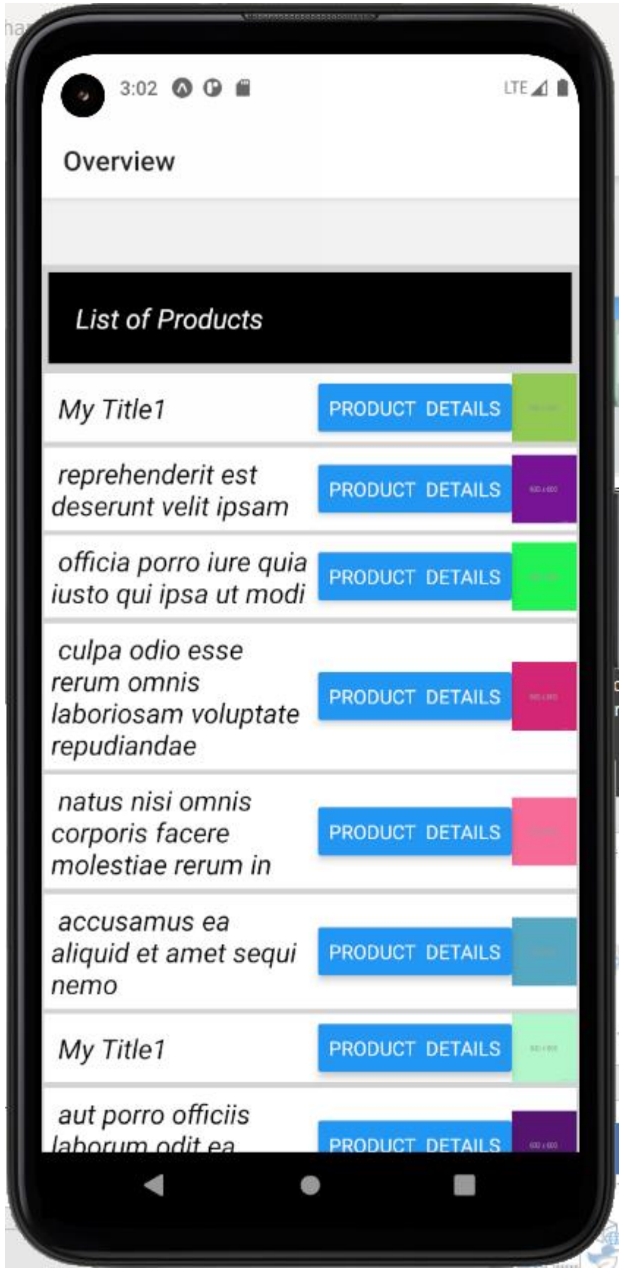
Name	Type	Description
albumId	123 number	Created by import
id	123 number	Created by import
title	A text	Created by import
url	A text	Created by import

```
[
 {"_id":"61744d8d868101090007b477",
"albumId":1,
"id":3,
"title":"ezz",
"url":"https://via.placeholder.com/600/24f355",
"thumbnailUrl":"https://via.placeholder.com/150/24f355",
"name":"pro1"},
 {"_id":"61744d8d868101090007b478",
"albumId":1,"id":4,
"title":"My Title4",
"url":"https://via.placeholder.com/600/d32776",
"thumbnailUrl":"https://via.placeholder.com/150/d32776",
"name":"pro4"}
]
```

_id is the object identifier of any json object stored in restdb

Change the Product List Screen

```
<FlatList data={data}
     keyExtractor={(item, index) => index.toString()}
     renderItem={({ item }) => (
       <View style={styles.row}>
         <Text style={styles.text}> {item['title']} </Text>
         <Button title="Product  Details"
           onPress={() => navigation.navigate('Details',
                                             { productID: item['_id']
                                              })
                   }
            />
            <Image style={styles.tinyLogo} source={{ uri: item['url']}}
            />
          </View>
        )} >
      </FlatList>
```



List of Products

My Title1

reprehenderit est deserunt velit ipsam

officia porro iure quia iusto qui ipsa ut modi

culpa odio esse rerum omnis laboriosam voluptate repudiandae

natus nisi omnis corporis facere molestiae rerum in

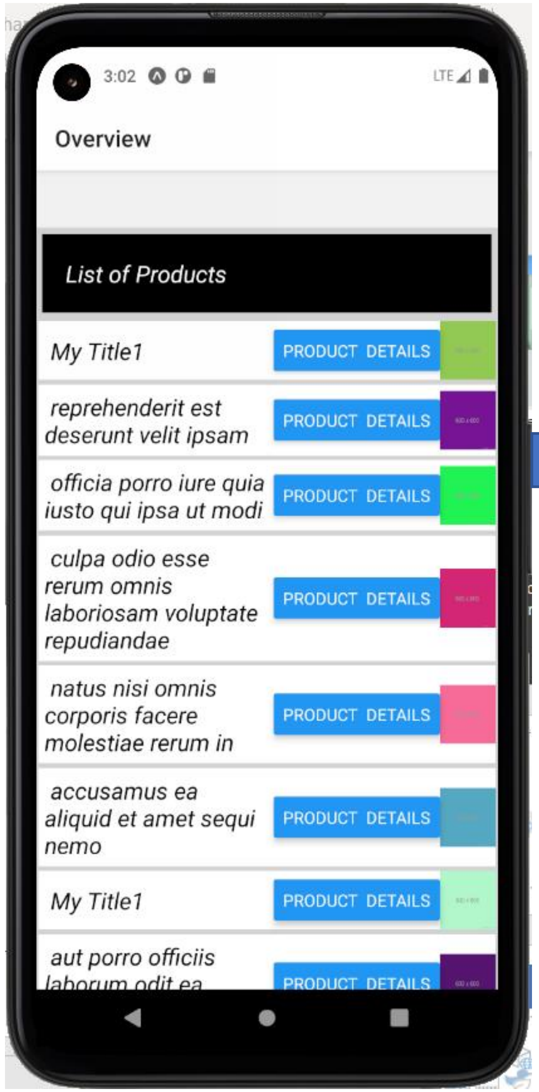
accusamus ea aliquid et amet sequi nemo

My Title1

aut porro officiis laborum odit ea

Navigation to Detail Screen

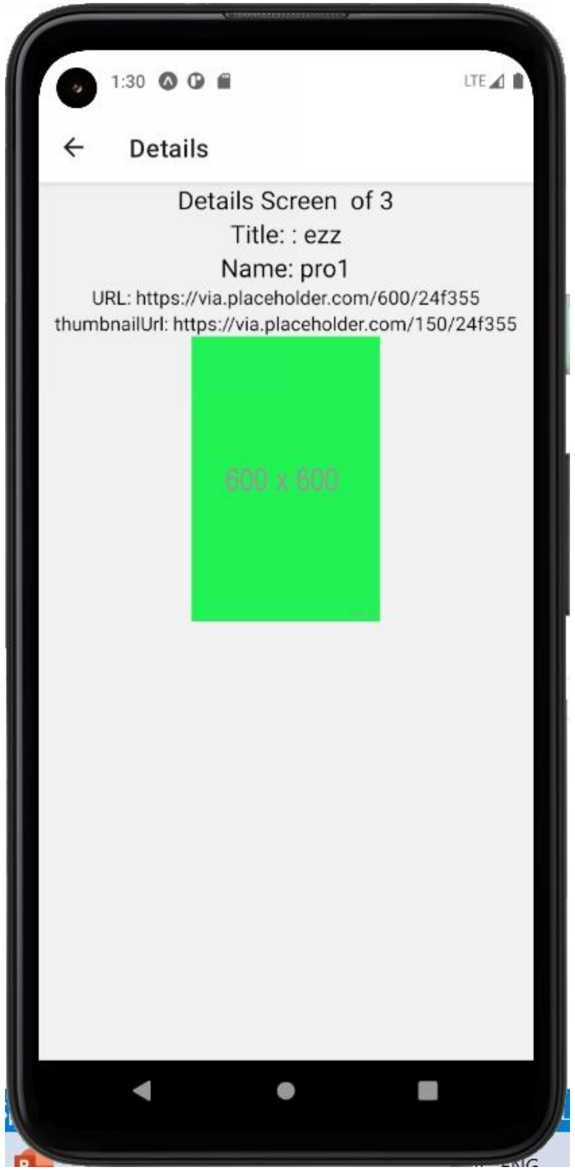
- Add Detail screen to retrieve detail of product using product ID (ObjectID)



Overview

List of Products

My Title1	PRODUCT DETAILS
reprehenderit est deserunt velit ipsam	PRODUCT DETAILS
officia porro iure quia iusto qui ipsa ut modi	PRODUCT DETAILS
culpa odio esse rerum omnis laboriosam voluptate repudiandae	PRODUCT DETAILS
natus nisi omnis corporis facere molestiae rerum in	PRODUCT DETAILS
accusamus ea aliquid et amet sequi nemo	PRODUCT DETAILS
My Title1	PRODUCT DETAILS
aut porro officiis laborum odit ea	PRODUCT DETAILS



← Details

Details Screen of 3

Title: : ezz

Name: pro1

URL: https://via.placeholder.com/600/24f355

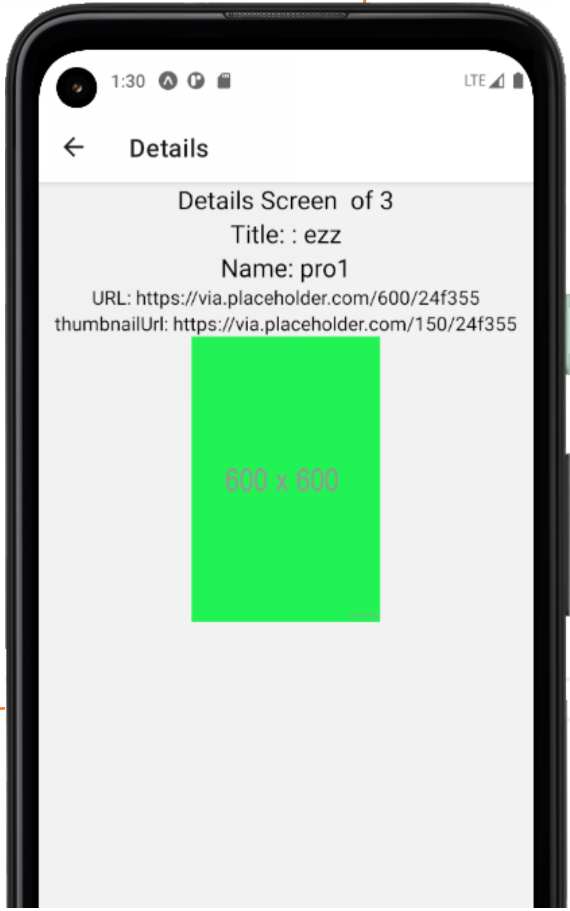
thumbnailUrl: https://via.placeholder.com/150/24f355

Fetch one Product details

```
function DetailsScreen({ route, navigation }) {
  /* 2. Get the param */
  const { productID } = route.params;
  const [productData, setproductData] = useState('');
  useEffect(() => {
    var options = {
      method: 'GET',
      headers:
      {
        'cache-control': 'no-cache',
        'x-apikey': 'e10a33c60bc1268c84e48cada806009efe740'
      }
    };
    fetch('https://cis424-ab14.restdb.io/rest/products/' + productID, options)
      .then((response) => response.json())
      .then((responseJson) => {
        console.log(responseJson);
        console.log(Date());
        setproductData(responseJson);
      })
      .catch((error) => {
        console.error(error);
      });

  }
, []); //useeffect end brackets add [] to avoid
```

Fetch one Product details

```
return (
    <View style={{ flex: 0.5, alignItems: 'center', justifyContent: 'space-evenly' }}>
      <Text>Details Screen   {productData.id} </Text>
      <Text>catName: {productData.title}</Text>
     <Text>catName: {productData.name}</Text>
     <Text>catName: {productData.url}</Text>
     <Text>catName: {productData.thumbnailUrl}</Text>
     <Image
                  style={styles.largeLogo}
                  source={{ uri: productData.url }}
                /> 

    </View>
  );
}
```



Details Screen of 3

Title: : ezz

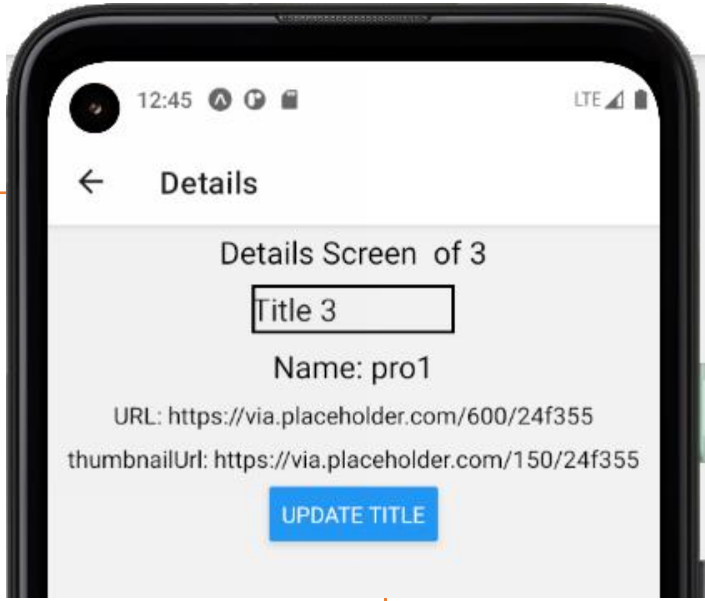
Name: pro1

URL: https://via.placeholder.com/600/24f355

thumbnailUrl: https://via.placeholder.com/150/24f355

Update Product details

```
function DetailsScreen({ route, navigation }) {
  /* 2. Get the param */
  const { productID } = route.params;
  const [productData, setproductData] = useState(‘’);
  const [productTitle, setproductTitle] = useState();
….
…..
    <Text style={{ fontSize: 20 }} > Title: {productData.title} </Text>
      <TextInput style={{ borderWidth: 2, width: 130, fontSize: 20 }}
        //value={productData.title}
        defaultValue={productData.title}
        onChangeText={setproductTitle}
      />
…
```



Details

Details Screen of 3

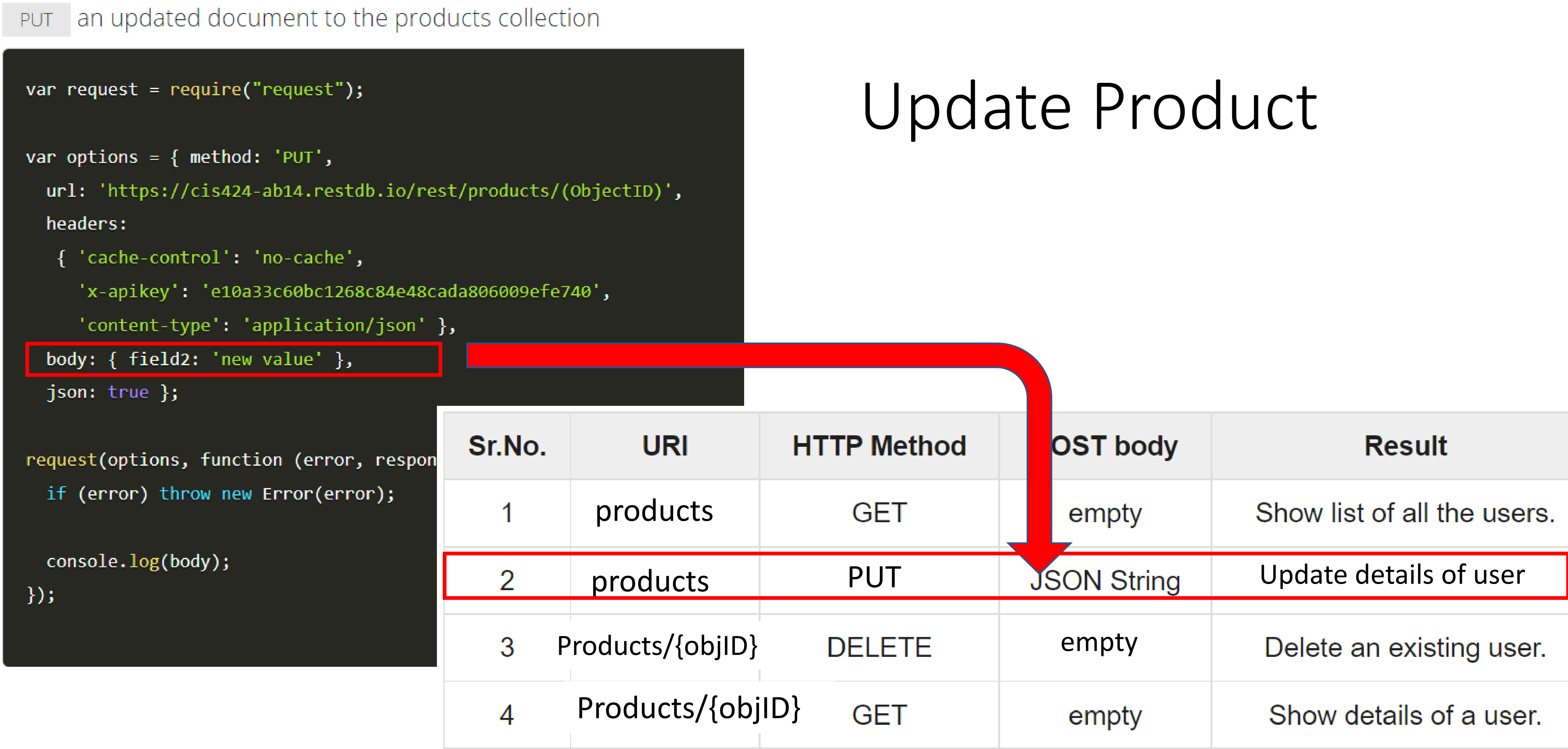
Title 3

Name: pro1

URL: https://via.placeholder.com/600/24f355

thumbnailUrl: https://via.placeholder.com/150/24f355

Update Product



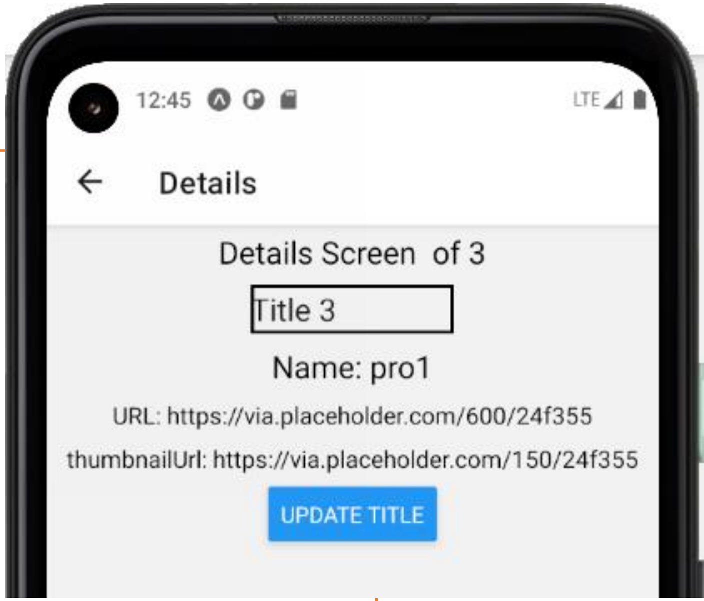
PUT an updated document to the products collection

```
var request = require("request");

var options = { method: 'PUT',
  url: 'https://cis424-ab14.restdb.io/rest/products/(ObjectID)',
  headers:
   { 'cache-control': 'no-cache',
     'x-apikey': 'e10a33c60bc1268c84e48cada806009efe740',
     'content-type': 'application/json' },
  body: { field2: 'new value' },
  json: true };

request(options, function (error, respon
  if (error) throw new Error(error);

  console.log(body);
});
```

Sr.No.	URI	HTTP Method	OST body	Result
1	products	GET	empty	Show list of all the users.
2	products	PUT	JSON String	Update details of user
3	Products/{objID}	DELETE	empty	Delete an existing user.
4	Products/{objID}	GET	empty	Show details of a user.

Update Product details

```
<Button title="Update Title"
    onPress={() => {
     var options = {
      method: 'PUT',
      headers:
      {'cache-control': 'no-cache',
       'x-apikey': 'e10a33c60bc1268c84e48cada806009efe740',
       'content-type': 'application/json'},
      body: JSON.stringify({
       title: productTitle  //add updated fields here
      }),
     json: true
     };
     fetch('https://cis424-ab14.restdb.io/rest/products/' + productID, options)
      .then((response) => response.json())
      .then((responseJson) => {
       console.log(responseJson);
       console.log(Date());
       setproductData(responseJson); // the object returned after updated
       navigation.navigate ('Home')
      })
      .catch((error) => {
       console.error(error);
      }); }}  />
```



Details Screen of 3

Title 3

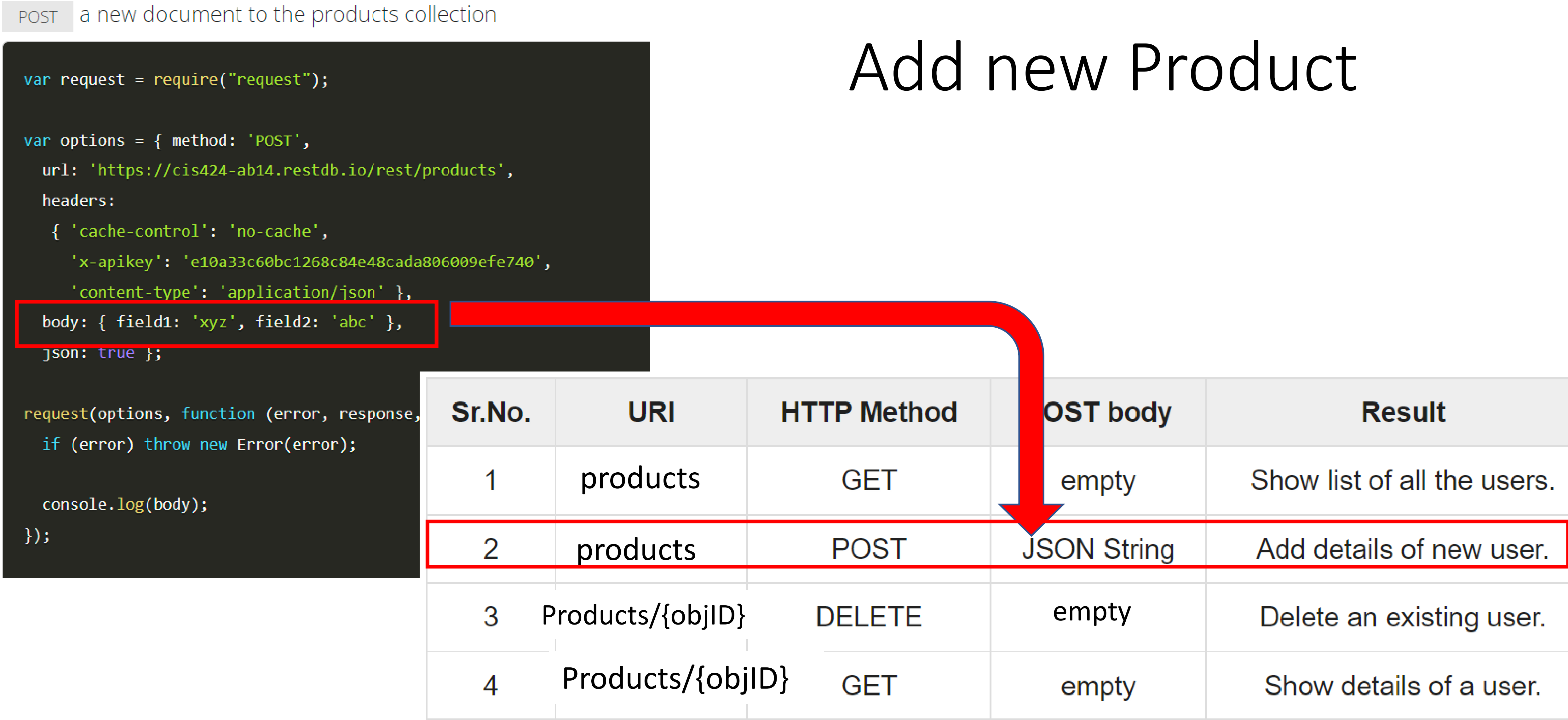
Name: pro1

URL: https://via.placeholder.com/600/24f355

thumbnailUrl: https://via.placeholder.com/150/24f355

Use replace to fix refresh issue

Add new Product



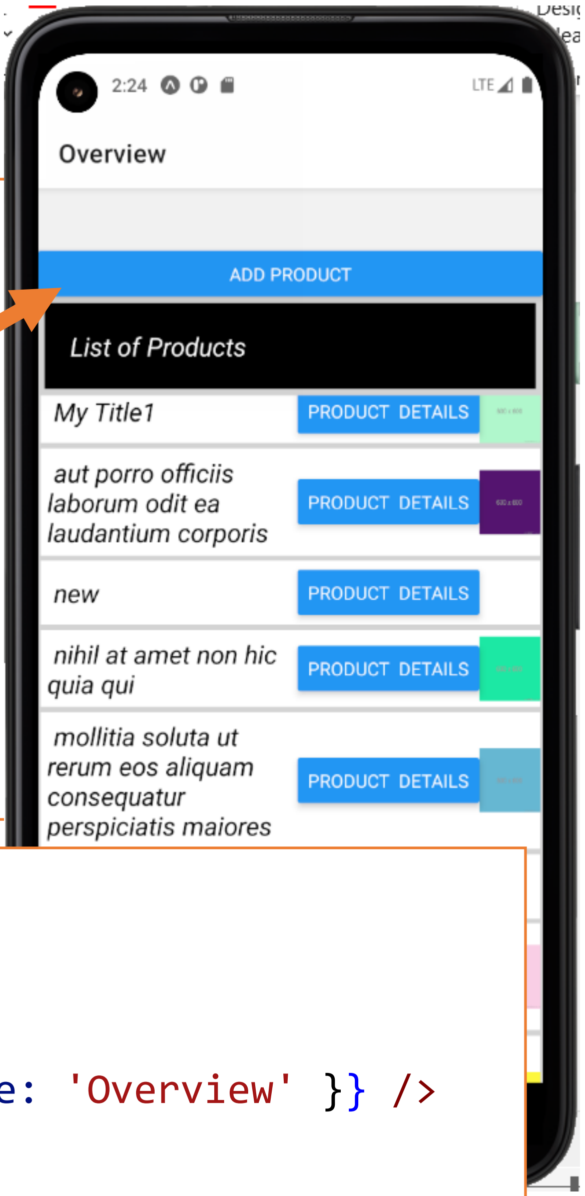
POST a new document to the products collection

```
var request = require("request");

var options = { method: 'POST',
  url: 'https://cis424-ab14.restdb.io/rest/products',
  headers:
   { 'cache-control': 'no-cache',
     'x-apikey': 'e10a33c60bc1268c84e48cada806009efe740',
     'content-type': 'application/json' },
  body: { field1: 'xyz', field2: 'abc' },
  json: true };

request(options, function (error, response,
  if (error) throw new Error(error);

  console.log(body);
});
```

Sr.No.	URI	HTTP Method	OST body	Result
1	products	GET	empty	Show list of all the users.
2	products	POST	JSON String	Add details of new user.
3	Products/{objID}	DELETE	empty	Delete an existing user.
4	Products/{objID}	GET	empty	Show details of a user.

Add Product

```
function ProductListScreen({ navigation }) {
….

return (
    <View style={styles.body}>
      <Button
        title="Add Product"
        onPress={() => navigation.navigate('NewProduct')}
      />
      <Text style={styles.header}>List of Products </Text>
      <FlatList data={data}
        keyExtractor={(item, index) => index.toString()}
```



2:24

Overview

ADD PRODUCT

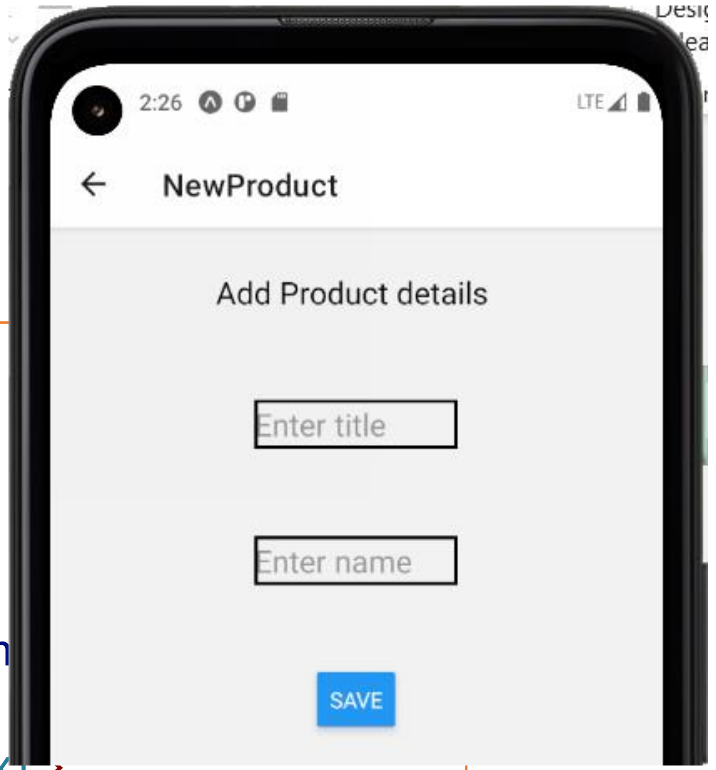
List of Products

My Title1	PRODUCT DETAILS
aut porro officiis laborum odit ea laudantium corporis	PRODUCT DETAILS
new	PRODUCT DETAILS
nihil at amet non hic quia qui	PRODUCT DETAILS
mollitia soluta ut rerum eos aliquam consequatur perspiciatis maiores	PRODUCT DETAILS

```
export default function App() {
  return (
    <NavigationContainer>
      <Stack.Navigator initialRouteName="Home">
        <Stack.Screen name="Home" component={ProductListScreen} options={{ title: 'Overview' }} />
        <Stack.Screen name="Details" component={DetailsScreen} />
        <Stack.Screen name="NewProduct" component={AddProductScreen} />
      </Stack.Navigator>
    </NavigationContainer>
  );
}
```

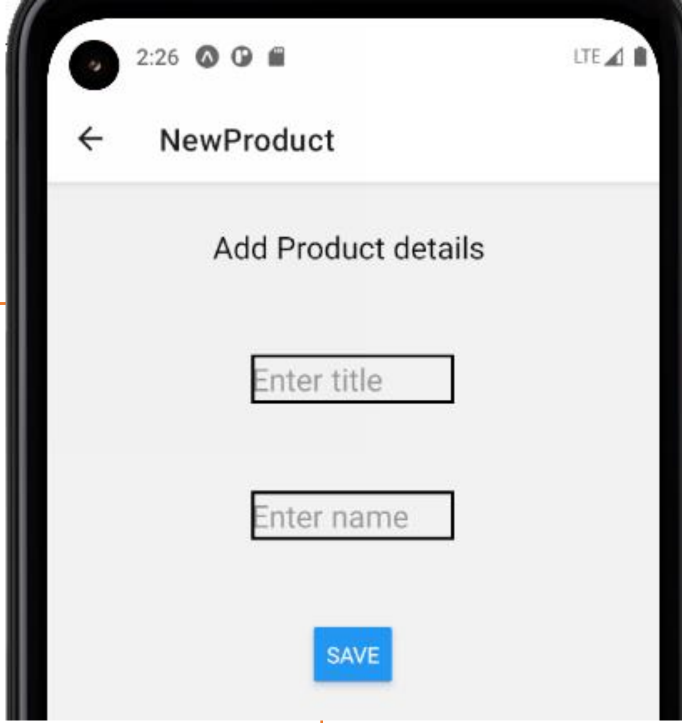
Add Product

```
function AddProductScreen({ route, navigation }) {
  const [productTitle, setproductTitle] = useState();
  const [productName, setproductName] = useState();
  return (
    <View style={{ flex: 0.5, alignItems: 'center', justifyConten
around' }}>
      <Text style={{ fontSize: 20 }} > Add Product details  </Text>
      <TextInput style={{ borderWidth: 2, width: 130, fontSize: 20 }}
        placeholder='Enter title'
        onChangeText={setproductTitle}
      />
      <TextInput style={{ borderWidth: 2, width: 130, fontSize: 20 }}
        placeholder='Enter name'
        onChangeText={setproductName}
      />
```



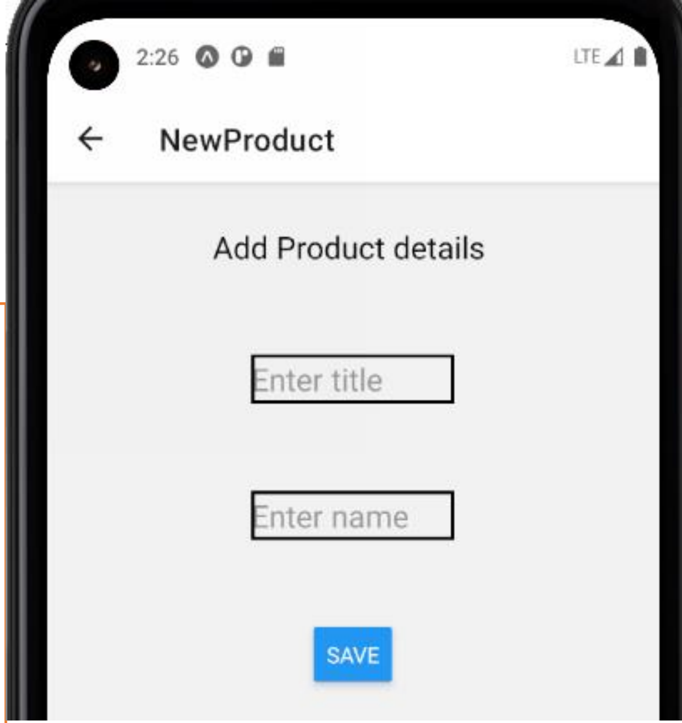
Add Product

```
  <Button
       title="Save"
       onPress={() => {
         if (productTitle == undefined || productName== undefined  )
           return
         var options = {
           method: 'POST',
           headers:
           {
             'cache-control': 'no-cache',
             'x-apikey': 'e10a33c60bc1268c84e48cada806009efe740',
             'content-type': 'application/json'
           },
           body: JSON.stringify({
             title: productTitle, name:productName
           }),
           json: true
         };
```

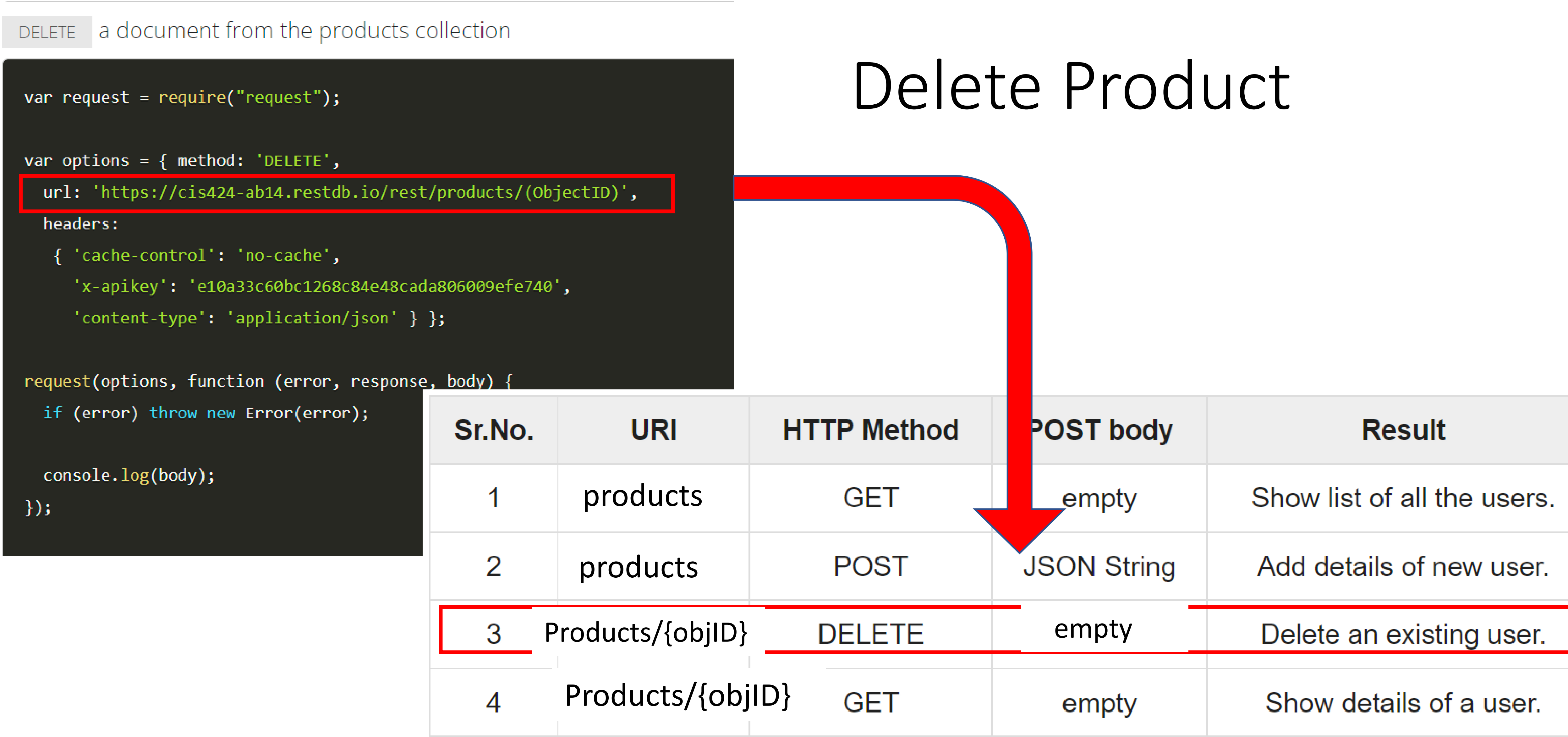


Add Product

```
   fetch('https://cis424-ab14.restdb.io/rest/products', options)
      .then((response) => response.json())
      .then((responseJson) => {
        console.log(responseJson);
        console.log(Date());
        navigation.replace('Home') //back to home
      })
      .catch((error) => {
        console.error(error);
      });
  }} //end of button
/>
</View>
);
```



Delete Product



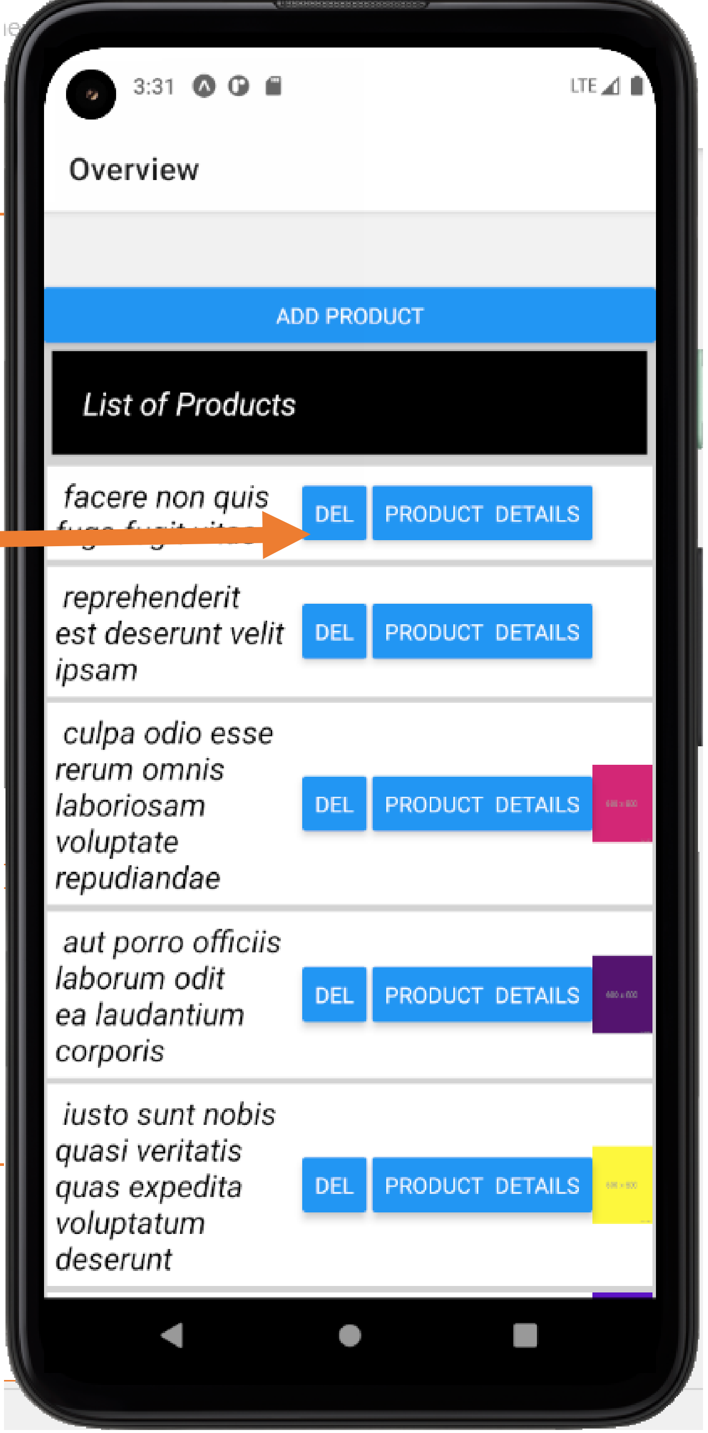
DELETE a document from the products collection

```
var request = require("request");

var options = { method: 'DELETE',
  url: 'https://cis424-ab14.restdb.io/rest/products/(ObjectID)',
  headers:
   { 'cache-control': 'no-cache',
     'x-apikey': 'e10a33c60bc1268c84e48cada806009efe740',
     'content-type': 'application/json' } };

request(options, function (error, response, body) {
  if (error) throw new Error(error);

  console.log(body);
});
```

Sr.No.	URI	HTTP Method	POST body	Result
1	products	GET	empty	Show list of all the users.
2	products	POST	JSON String	Add details of new user.
3	Products/{objID}	DELETE	empty	Delete an existing user.
4	Products/{objID}	GET	empty	Show details of a user.

Delete Product

```
function ProductListScreen({ navigation }) {
….

      <FlatList data={data}

<Button
        title="Del"
        onPress={() => {
           console.log('--Delete--' + item['_id'])
           var options = {
               method: 'DELETE',
               headers:
               {
                 'cache-control': 'no-cache',
                'x-apikey': 'e10a33c60bc1268c84e48cada806009efe740'

                 }
              };
```



3:31

Overview

List of Products

facere non quis

reprehenderit est deserunt velit ipsam

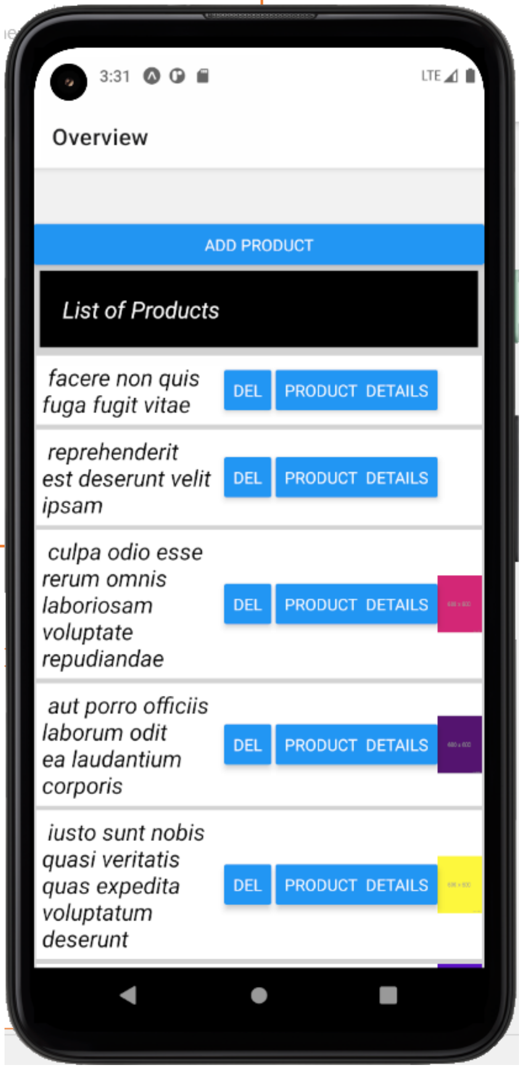
culpa odio esse rerum omnis laboriosam voluptate repudiandae

aut porro officiis laborum odit ea laudantium corporis

iusto sunt nobis quasi veritatis quas expedita voluptatum deserunt

Delete Product

```
fetch('https://cis424-ab14.restdb.io/rest/products/' + item['_id'], options)
       .then((response) => response.json())
       .then((responseJson) => {
         //console.log(responseJson);
         console.log("Deleted -- " + Date());
         navigation.replace('Home')
       })
       .catch((error) => {
         console.error(error);
       });
   }
   }
 />
```



facere non quis fuga fugit vitae

reprehenderit est deserunt velit ipsam

culpa odio esse rerum omnis laboriosam voluptate repudiandae

aut porro officiis laborum odit ea laudantium corporis

iusto sunt nobis quasi veritatis quas expedita voluptatum deserunt

Q& A