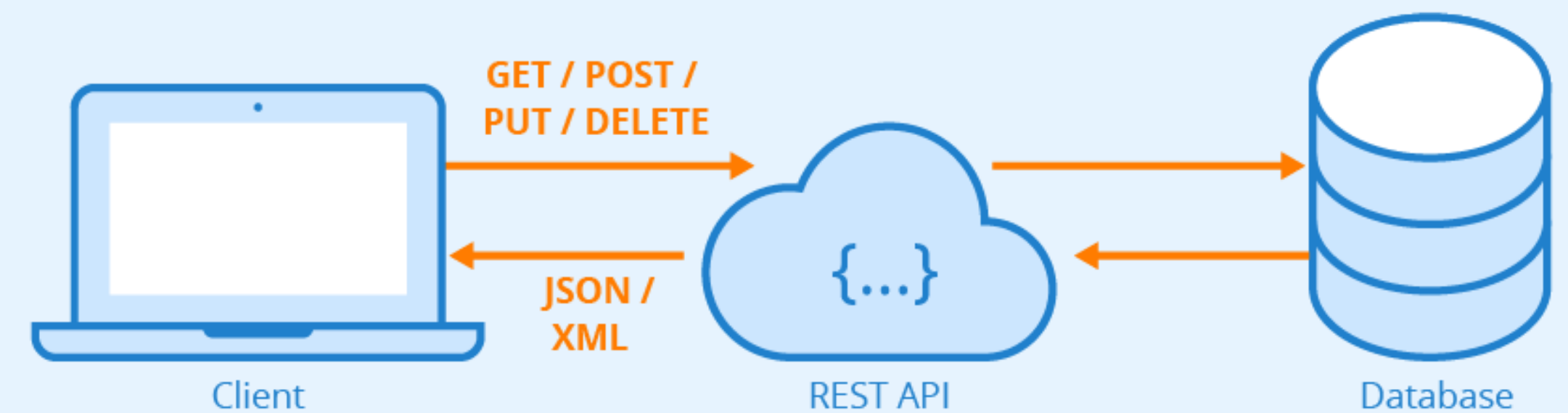


Express 2

What is a Restful API?

- API: an interface that sits between two applications or modules of code
- Restful API: a client/server architecture that uses a stateless communication protocol like HTTP
- RESTful web services are based on these principles:
- RESTful web services expose resources using URIs.
- Resources are manipulated using PUT, GET, POST, and DELETE operations.
- PUT creates a new resource, DELETE deletes a resource, GET retrieves the current state, POST transfers a new state onto a resource.



What is a Restful API?

- Resources are decoupled from their representation so that their content can be accessed in a variety of formats, such as HTML, XML, plain text, PDF, JPEG, JSON and others.
- Interaction with resources is stateless. State information is exchanged using techniques like URI rewriting, cookies, hidden form fields and embedding state information in response messages.

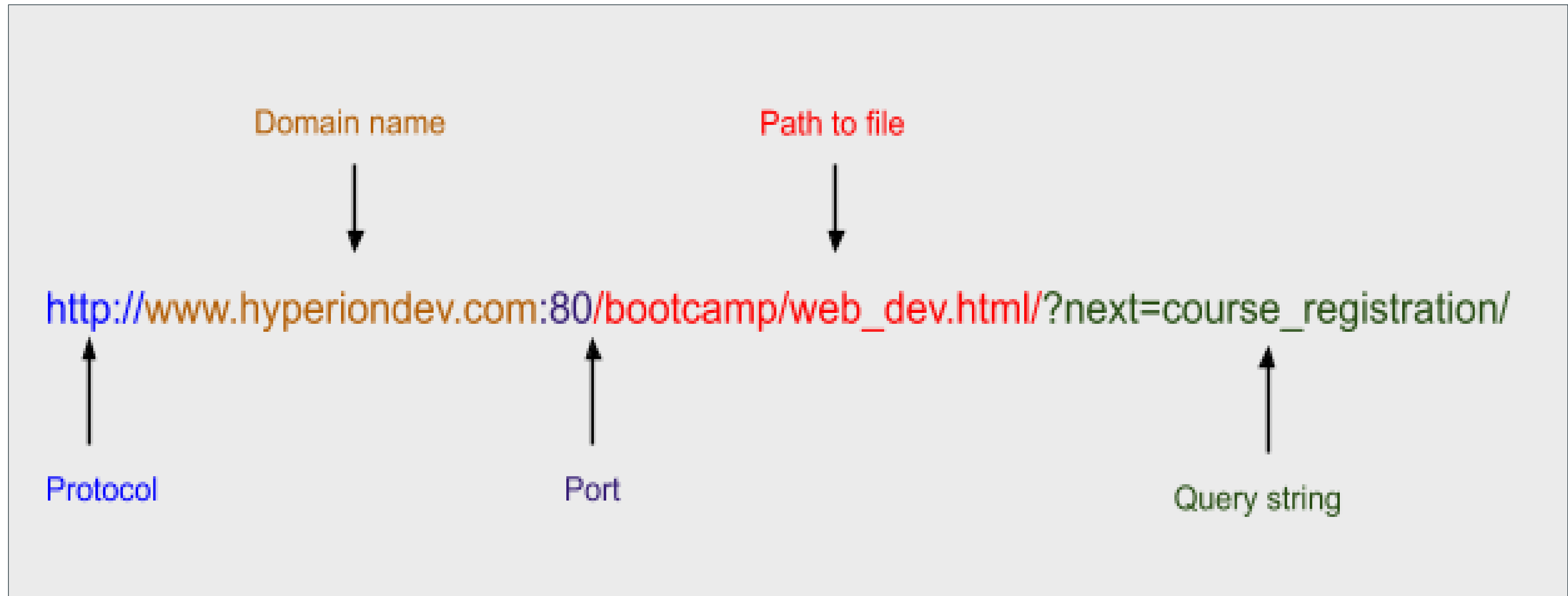
Create a Custom Restful API Using Express

- A Restful web API is code that is written to respond to HTML PUT, GET, POST and DELETE requests.
- To create a Restful API, we are going to write JavaScript functions using Express and Node to handle each of these requests.
- Like the `app.get()` method, each of these methods takes two arguments:
 - The route. These methods are used to perform routing.
 - A callback function.
- Each route handler that we write will be used to either create (e.g. create a JSON file), read, update or delete data (CRUD)

Create a Custom Restful API Using Express

HTTP verb	CRUD operation	Express method	Description
Post	Create	app.post()	Used to submit some data about a specific entity to the server.
Get	Read	app.get()	Used to get a specific resource from the server.
Put	Update	app.put()	Used to update a piece of data about a specific object on the server.
Delete	Delete	app.delete()	Used to delete a specific object.

Passing Data Through to the Server Using the Request Object



Passing Data Through to the Server Using the Request Object

`http://www.hyperiondev.com:80/portal/2315/`



Parameter

Passing Data Through to the Server Using the Request Object

- To access the data passed through using the URL, we use the req object that is passed through as an argument to the app.post or app.put route handler.

```
app.post('/', (req, res) => {  
  fileHandler.writeFile('person.json', `{name: ${req.query.name}}`, (err) => {  
    if (err) throw err;  
    res.send('File created!');  
  });  
});
```

URL: localhost:3000?name=Gareth



Passing Data Through to the Server Using the Request Object

- See in the example below how the code `req.params.name` is used to get the value of the parameter 'name' that is defined in the route argument of the `app.put()` method.

```
app.put('/:name', (req, res) => {  
  fileHandler.writeFile('person.json', `{name: ${req.params.name}}`, (err) => {  
    if (err) throw err;  
    res.send('File updated!');  
  });  
});
```

URL: localhost:3000/Sue

Postman

- Postman: a free API development environment.
- Postman: It presents you with a friendly GUI for constructing requests and reading responses.
- How to install: Download it from their website and install postman on your machine.

Test API in next slides using Postman 🖥️ 📱



Working with Requests

- Write a Get Method to get a files contents when you hit an end point as follows.

```
const bodyParser = require('body-parser');
app.use(bodyParser.urlencoded({ extended: true }))
app.use(bodyParser.json())

app.get('/name', function(req, res) {
  console.log('App.get');
  fileHandler.readFile('person.json', (err, data) => {
    if (err) res.json({"message": 'File not found. First post to create file.'});
    else
      res.json({"message": `Hello World! ${data}`});
  })
})
```

Working with Requests

- Write a POST Method to add to a file like this.

```
app.post('/name', (req, res) => {  
  console.log('App.post');  
  console.log(req.body.name);  
  fileHandler.writeFile('person.json', `{name: ${req.body.name}}`, (err) => {  
    if (err) throw err;  
    res.send({"message": "File created!"});  
  });  
})
```