

Assignment Project Exam Help  
REST

<https://tutorcs.com>

WeChat: cstutorcs  
Lecturer: Dr. Joseph Doyle

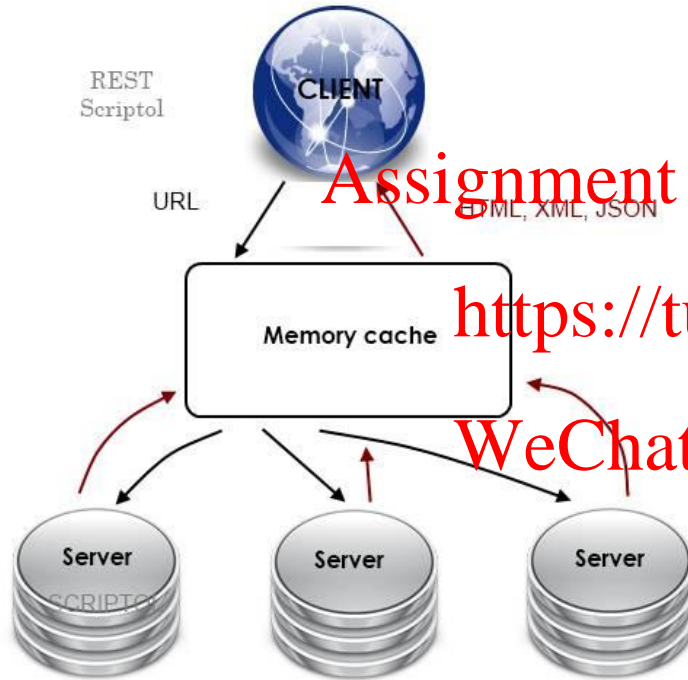
# Introduction

- REST stands for REpresentational State Transfer
- It was proposed by Roy Fieldings as part of his PhD thesis in 2000
- Important to note that REST is not a technology or framework.
- It is an “Architectural Style” which is defined as set of principles and constraints (In contrast SOAP is much more constrained)

# Properties

- A REST system should have the following properties:
  - It should implement the client-server architecture
  - It should be a stateless system meaning that each request should be treated independently and it should not require the storage of sessions for clients
  - It should support a caching system
  - It should be uniformly accessible meaning that each resource should have a unique address and a valid point of access

# REST Architecture



Could consider a web server as a REST system as when a browser enters a URL it receives a representation of the current state of that resource. The page may be relatively static such as documentation pages or relatively dynamic (News websites, blogs etc)

# REST Abstraction

- REST consists of three main abstractions
  - Resources
  - Representations
  - Actions

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# REST Resources

- A resource is anything that a service can provide
- States and functions are also considered to be resources
- Examples of resources include:
  - A review from Metacritic
  - An image from Imgur
  - A video from Youtube
  - Order information on Amazon

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# REST Resources

- A resource is defined as anything that can be accessed and transferred from a server to a client
- In order to be accessible a resource must have a unique address
- Each resource should have a unique Uniform Resource Identifier (URI) under HTTP (This is similar to SOAP. See previous slides)

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: estutorcs

# URI

- URIs are not intended to change over time
- It is the only method to locate a particular resource and if it is changed any links to it will also change
- The URI is also used to negotiate the particular representation of the resource that the client wants
- The client achieves this by providing parameters in the URI to define what information the client wants
- The server responds with a resource representation containing the information the client asked for (assuming that the client has access)



# Structure of URI

- URI = scheme[userinfo@]host[:port]path[?query][#fragment]
- <https://john.doe@www.example.com:23/forp/questions/?tag=networking&order=newest#top>
- In this case ?query allows the client to pass parameters to the server to identify the particular resource that the client desires

# Representations

- Representations of resources are the items that are transferred between clients and servers
- Resources are never sent only their representations
- The format of the representation is determined by its content-type
- The interaction of the representation of the resource is determined by the action

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Representations

- REST uses HTTP for communication so any information that can be passed between clients and servers can be transferred
- If we assume that the data is transferred over the TCP/IP protocol the browser knows how to interpret the idea as this information is detailed in HTTP response head Content Type
- Different clients are able to consume different representations of the same resource
- The representation can be in text, image, video or JSON format but it has to be available through the same URI

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutores

# Representations

- Examples of different representation types being sent include:
  - Sending data as JSON rather than as plain text as readability is not important
  - Sending a video with a resolution related to the platform. E.g. if a mobile device can only support 720p video do not send it 1080p video (This of course assumes that the client is clever enough to include this information in the parameters based to the REST server)

# Actions

- Actions are used to operate on resources
- They include:
  - A reviews related to a particular film on Metacritic
  - Posting an image on Imgur
  - A video from Youtube
  - Deleting a file from a folder on Dropbox

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# HTTP Actions

- Under HTTP the standard actions:
  - GET [Assignment Project Exam Help](#)
  - POST <https://tutorcs.com>
  - PUT [WeChat: cstutorcs](#)
  - DELETE

# Data Actions

Data Action	HTTP Protocol Equivalent
CREATE	POST
RETRIEVE	GET
UPDATE	PUT
DELETE	DELETE

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# HTTP as Uniform Interface

- In RESTful system there is an increased focus on resource names
- RESTful systems have four specific actions that can interact with resources – Create, Retrieve, Update and Delete (CRUD)
- This contrasts with other web applications which have no naming or implementation standards

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



# Example 1 Instagram

- Consider the Instagram API
- Service is located at <https://api.instagram.com>
- The resources are represented as JSON files
- The URI of a user identified by a user-id is <https://api.instagram.com/v1/users/{user-id}>
- The URI of a search function for users is <https://api.instagram.com/v1/users/search>
- The URI of a location identified by a location-id is <https://api.instagram.com/v1/locations/{location-id}>

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Example 1 Instagram (cont)

- If we wanted to find photo locations related to a set of longitude and latitude coordinates (40.7127° N, 74.0059 °W) we could use  
[http://api.instagram.com/v1/locations/search?access\\_token=ACCESS\\_TOKEN&lat=40.7127&lng=74.0059](http://api.instagram.com/v1/locations/search?access_token=ACCESS_TOKEN&lat=40.7127&lng=74.0059)
- This would return a JSON response depicted on the next slide

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Example 1 Instagram (cont)

HTTP/1.1 200 OK

```
{
  "meta": {
    "code": 200
  },
  "data": [
    {
      "latitude": 40.714198749,
      "id": "93496093",
      "longitude": 74.006001183,
      "name": "John's Pizzeria 278 Bleecker St NY, NY"
    },
    {
      "latitude": 40.7142,
      "id": "46371155",
      "longitude": 74.0064,
      "name": "Thunderpocalypse 2012"
    }
  ]
}
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

<https://openclassrooms.com/en/courses/3432056-build-your-web-projects-with-rest-apis/3496011-examples-of-rest-apis>

# Example 2 Github

- Consider the Github API
- Service is located at <https://api.github.com>
- The resources are represented as JSON files
- The URI of a user identified by a user-id is <https://api.github.com/users/{user-id}>
- The URI of an organisation identified by an organisation id is <https://api.github.com/orgs/{organisation-id}>
- The URI of a repository identified by a repository-id is <https://api.github.com/repos/{repository-id}>

Assignment Project Exam Help

<https://tutors.com>

WeChat: estutors

# Example 2 Github (cont)

- If we wanted to find the information about the Github user “octocat” we could use <https://api.github.com/users/octocat>
- This would return a JSON response depicted on the next slide

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Example 2 Github (cont)

```
{
  "login": "octocat",
  "id": 1,
  "avatar_url": "https://github.com/images/error/octocat_happy.gif",
  "gravatar_id": "",
  "url": "https://api.github.com/users/octocat",
  "html_url": "https://github.com/octocat",
  "followers_url": "https://api.github.com/users/octocat/followers",
  "following_url": "https://api.github.com/users/octocat/following{/other_user}",
  "gists_url": "https://api.github.com/users/octocat/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/octocat/starred{/owner}/{repo}",
  "subscriptions_url": "https://api.github.com/users/octocat/subscriptions",
  "organizations_url": "https://api.github.com/users/octocat/orgs",
  "repos_url": "https://api.github.com/users/octocat/repos",
  "events_url": "https://api.github.com/users/octocat/events{/privacy}",
  "received_events_url": "https://api.github.com/users/octocat/received_events",
  "type": "User",
  "site_admin": false,
  "name": "monalisa octocat",
  "company": "GitHub",
  ...
}
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# GET

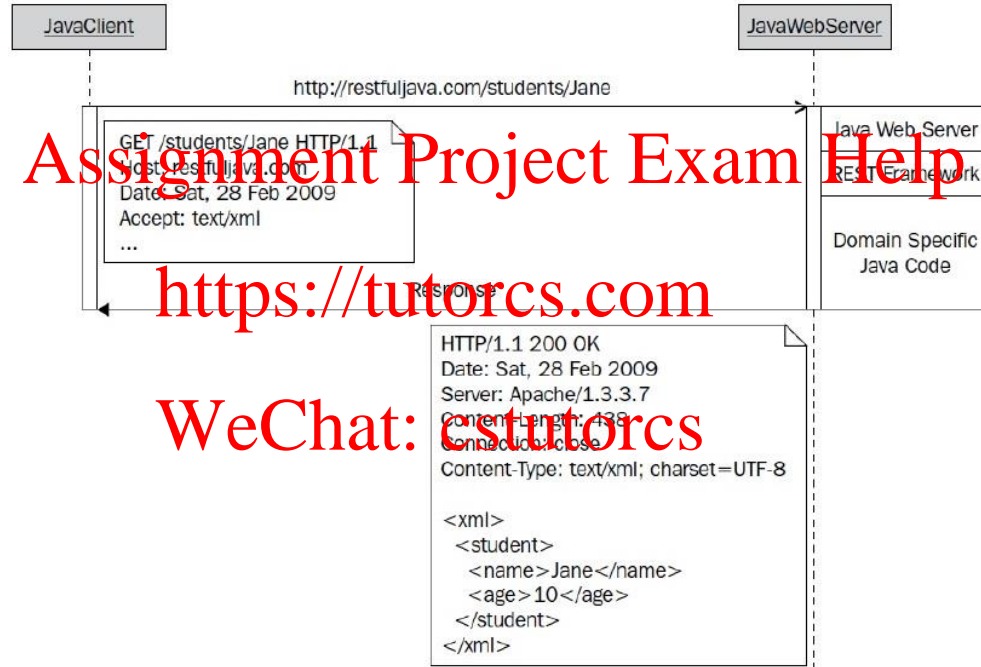
- The GET method is used to RETRIEVE resources
- It should not be able to affect a resource
- Multiple GETs can be called without changing the state of the resource
- It can also return parts of a resource
- By altering the URI it can act to read the resource and as a query operation

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# GET Example



<http://www.restfuljava.com>



# POST

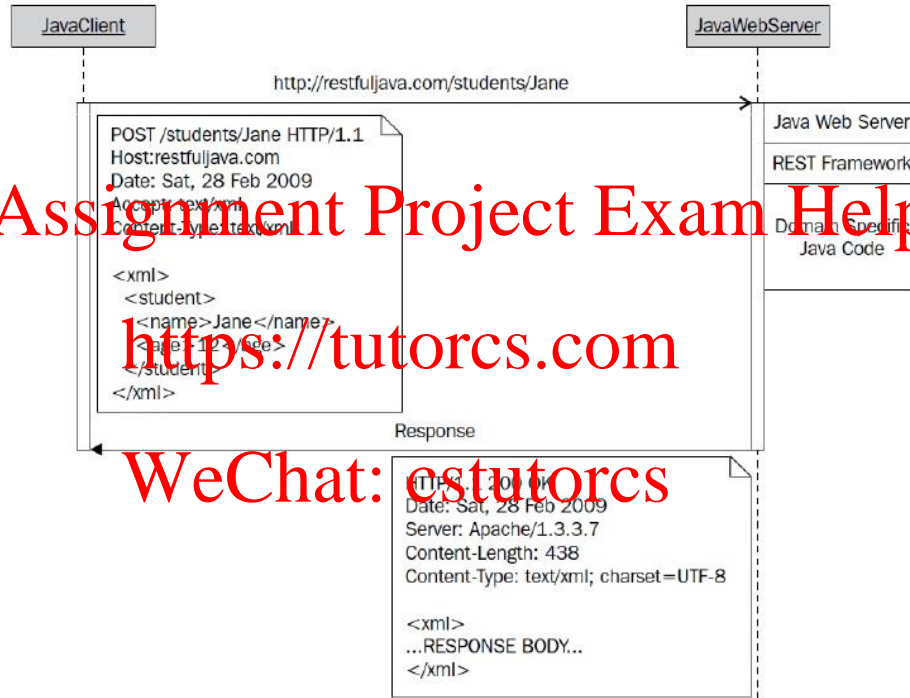
- The POST method is used to CREATE resources
- The resource URI is usually not known at creation time
- The REST server usually automatically generates the new URI for the resource

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# POST Example



<http://www.restfuljava.com>

# PUT Example

- The PUT method is used to UPDATE resources
- It works as follows:
  - A GET request is used to obtain a representation of the resource which needs to be updated
  - The client updates the resource with new values
  - The resource is updated using a PUT request with the representation as a payload

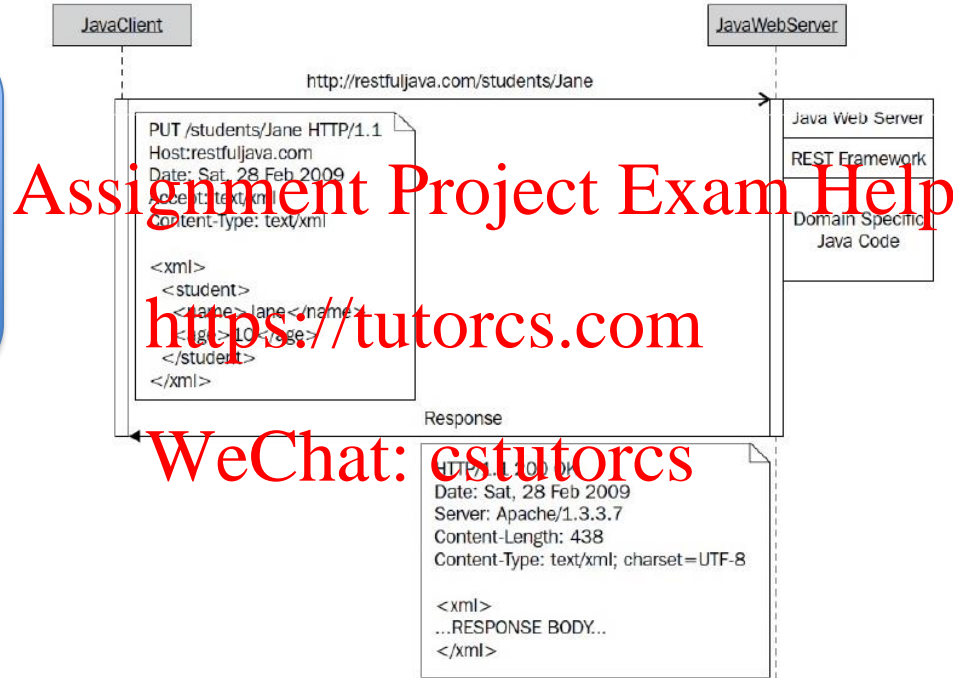
Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# PUT Example

The initial GET request is omitted in this diagram

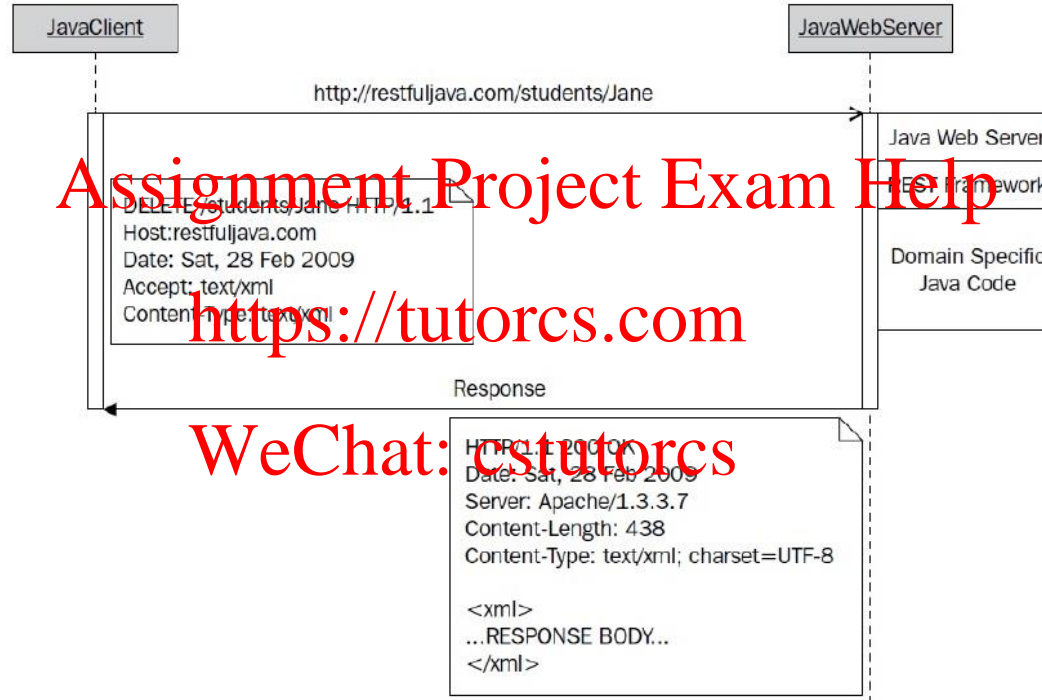


<http://www.restfuljava.com>

# DELETE Example

- The DELETE method is used to DELETE resources
- It works in a similar fashion to PUT
  - Both PUT and DELETE requests apply to the entire resource
  - Both PUT and DELETE requests are atomic so two requests so simultaneous requests are managed and one request will determine the final state of the resource
- It can also be used to determine the URI of the resource which is being deleted

# DELETE Example



<http://www.restfuljava.com>

# HTTP Status Codes

- RESTful services use HTTP status codes to return information about the responses to requests
- Broadly these codes fall into the following categories
  - 1XX Informational Message
  - 2XX Success Message
  - 3XX Redirect the client to another URL
  - 4XX Client-side error
  - 5XX Server-side error

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Security

- There are three main methods for securing user access to RESTful services namely:
  - Custom token authentication
  - HTTP Basic authentication
  - OAuth
- OAuth is used to access service on behalf of users while the other methods control access to the resources

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



# Custom Token Authentication

- Custom Token authentication functions as follows:
  - A unique token is generated for a registered API user by a server
  - This token is sent by the register user with every request to the service
  - The token can be sent as part of the URL or it can added to the HTTP request header
- This can be used for additional functionality
- For example if charging was based upon the number of GET requests (AWS S3) this could be used to generate bills

# Advantages and Disadvantages

- Advantages

- The generation of the token is independent of the web service
- It is simple to implement
- It is possible to log data exchange

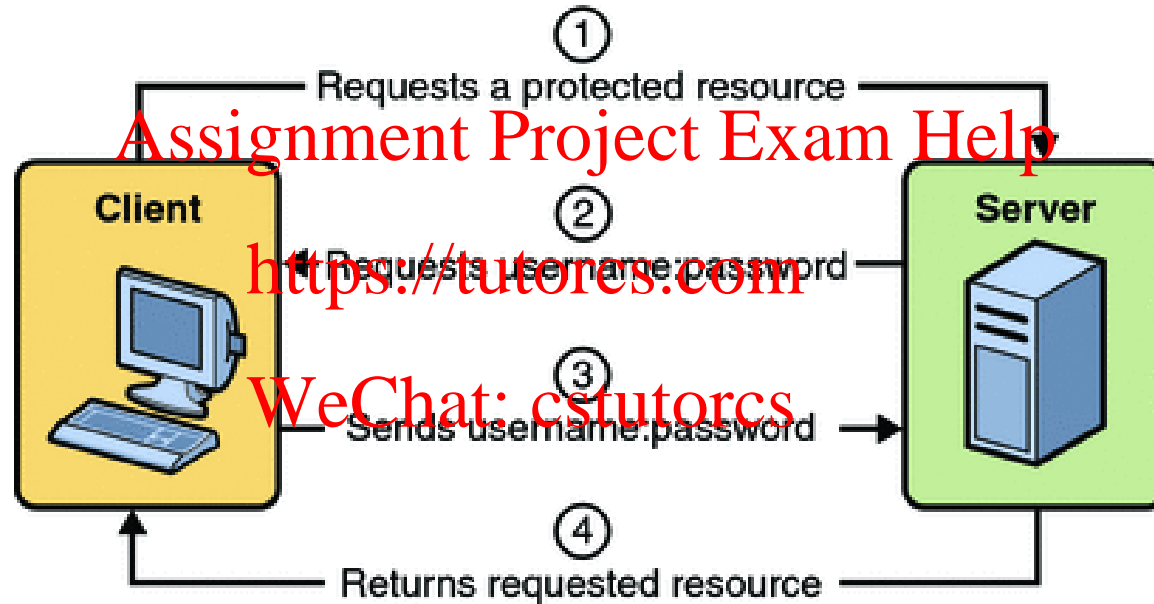
- Disadvantages

- The method is not secure as the token can be copied and reused without authorization

# HTTP Basic Authentication

- With this method the client sends the username and password pair in the HTTP header Authorization
- The username and password pair is in cleartext Base64 encoded. This can take two forms:
  - A client makes a request without authentication credentials and receives a HTTP error code 401 which must be handled by the client
  - The client includes the credentials with the initial request
- This username and password must be sent with every HTTP request so that the authorization is validated

# HTTP Basic Authentication (cont)



<https://docs.oracle.com/javaee/5/tutorial/doc/bncbe.html>

# Advantages and Disadvantages

- Advantages

- The client can manage server authorization requests
- This method is reasonably secure if HTTPS(SSL) is used rather than HTTP

- Disadvantages

- This method with HTTP only is not secure as Base64 encoding can easily be deciphered

# OAuth 2.0

- This is the preferred authorization scheme for RESTful services
- It is easy to integrate and an open source protocol
- OAuth allows third party to access user's data stored in a web service
- This occurs with the users' consent and does not require a username and password exchange

Assignment Project Exam Help

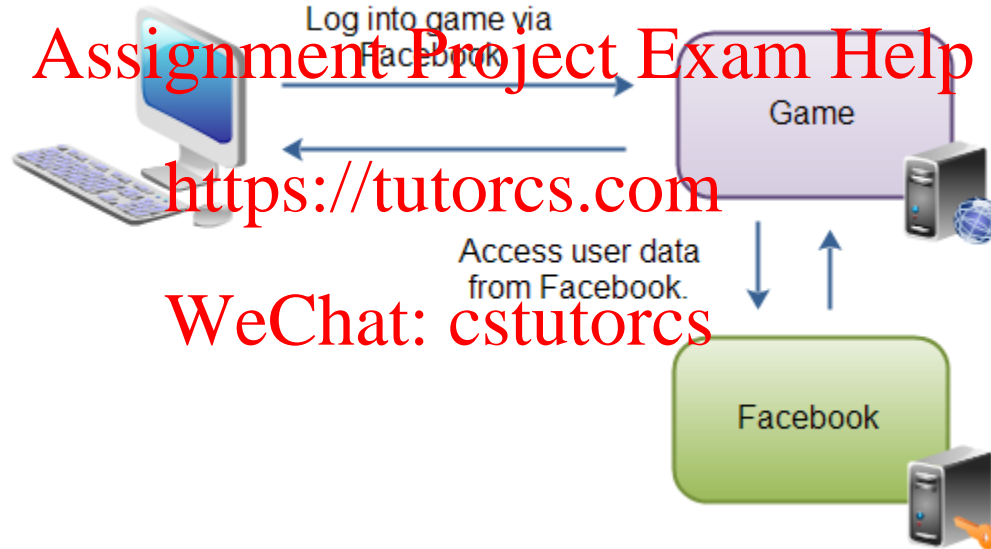
<https://tutorcs.com>

WeChat: cstutorcs

# OAuth 2.0 (cont)

- Users do not provide login details to third party services [Assignment Project Exam Help](#)
- Authorization is granted by the web service which stores the user's data and credentials <https://tutorcs.com>
- The third party only receives an authorization token which can be used to access the data from the web service [WeChat: cstutorcs](#)

# OAuth 2.0 Example



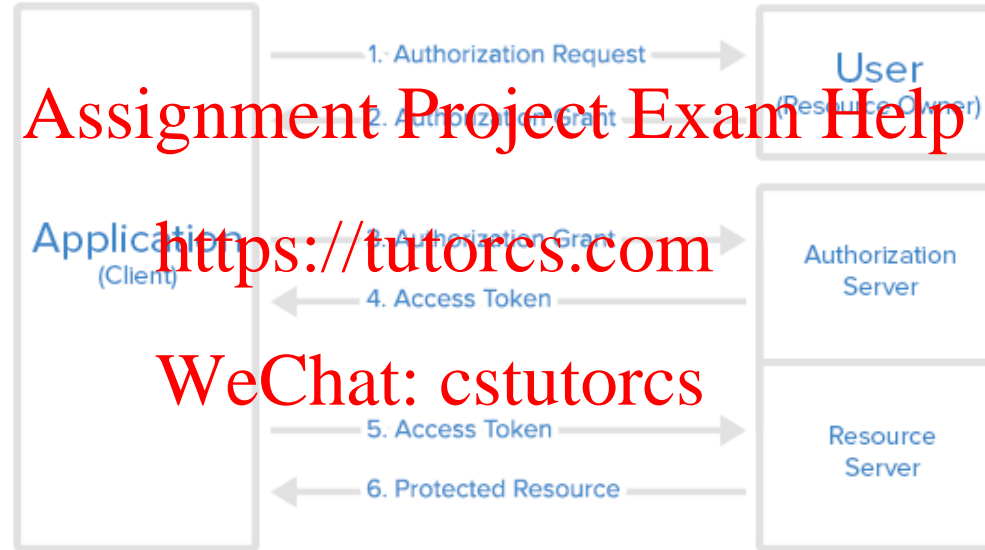
<http://tutorials.jenkov.com/oauth2/index.html>



# OAuth 2.0 Example (cont)

- The game needs authentication to access part of a website
  - Post a message on a users wall Facebook
  - Inform the user which of its friends are also playing the game
- The game has a unique token which is used to access this information

# OAuth 2.0 Process



<https://www.digitalocean.com/community/tutorials/an-introduction-to-oauth-2>

# Java Rest Example

```
package com.example.rests-service;
```

```
public class Greeting {
```

```
    private final long id;
```

```
    private final String content;
```

```
    public Greeting(long id, String content) {
```

```
        this.id = id;
```

```
        this.content = content;
```

```
    }
```

```
    public long getId() {
```

```
        return id;
```

```
    }
```

```
    public String getContent() {
```

```
        return content;
```

```
    }
```

```
}
```

<https://spring.io/guides/gs/rest-service>

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# Java Rest Example (cont)

```
package com.example.restservlet;  
import java.util.concurrent.atomic.AtomicLong;  
import org.springframework.web.bind.annotation.GetMapping;  
import org.springframework.web.bind.annotation.RequestParam;  
import org.springframework.web.bind.annotation.RestController;  
@RestController  
public class GreetingController {  
    private static final String template = "Hello, %s!";  
    private final AtomicLong counter = new AtomicLong();  
    @GetMapping("/greeting")  
    public Greeting greeting(@RequestParam(value = "name", defaultValue = "World") String name) {  
        return new Greeting(counter.incrementAndGet(), String.format(template, name));  
    }  
}
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# GraphQL (Successor to REST?)

- GraphQL is an open-source data query and manipulation language, and a runtime for fulfilling queries with existing data.
- Developed internally in Facebook in 2012
- Publicly released in 2015
- Clients can define the structure of the data required, and exactly the same structure of the data is returned from the server
- In theory, this could reduce the data being transferred

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# GraphQL Design

- GraphQL models data as types and fields on those types
- Functions are then created for each field on each type
- When the GraphQL function is running it can be sent queries which validated by ensuring that it only refers to the fields and types defined
- Once this is successfully completed the functions are used to produce a result which can be in different formats e.g. JSON, XML etc.
- Mutations can be defined which are used to update or create data

Assignment Project Exam Help

<https://tutorso.com>

WeChat: cstutorcs

# GraphQL Example Query

```
{  
  hero {  
    name  
  }  
}
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

<https://graphql.github.io>

# GraphQL Example Query

```
{  
  "data": {  
    "hero": {  
      "name": "R2-D2"  
    }  
  }  
}
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs



# GraphQL Example Mutation

mutation CreateReviewForEpisode(\$ep: Episode!, \$review:  
ReviewInput!)

**Assignment Project Exam Help**

createReview(episode: \$ep, review: \$review) {

**<https://tutorcs.com>**

stars

commentary

**WeChat: cstutorcs**

}

}

# GraphQL Example Mutation Variables

```
{  
  "ep": "JEDI",  
  "review": {  
    "stars": 5,  
    "commentary": "This is a great movie!"  
  }  
}
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# GraphQL Example Mutation Result

```
{  
  "data": {  
    "createReview": {  
      "stars": 5,  
      "commentary": "This is a great movie!"  
    }  
  }  
}
```

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# GraphQL Vs REST

- GraphQL provides reflective discoverability which allows clients to inspect the schema of a GraphQL server but is only related to the schema and not the stored data
- REST systems which use hypermedia controls have dynamic discoverability which can greatly simplify clients as information for the processing of media does not have to be stored with the client
- GraphQL is a solution to a very specific problem which makes it simple to use
- REST has resulted in a lot of standardisation which makes it easy to use as many libraries and existing tools can be used in the creation of application
- GraphQL uses the notion of reducing over-fetching and preventing multiple trips to improve performance
- REST uses caching to improve performance. Caching is not explicitly mentioned in GraphQL but technologies like Relay can be used to implement it with GraphQL services

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: cstutorcs

# GraphQL Conclusions

- In general GraphQL shifts a number of responsibilities from the server to client
- In addition, the general approach of GraphQL is return the smallest amount of data possible upon receipt of a query while the general approach of REST is to return as much data as possible
- It is possible to implement both (Github currently does this)
- REST implementations vary quite a bit and the appeal of the advertised benefits of GraphQL will vary significantly depending on the quality of the REST implementation

Assignment Project Exam Help

<https://tutorcs.com>

WeChat: estutorcs