[ZANG] - [Chuanjie]

Student ID: 699382953

[chuanjiezang@gmail.com](mailto:chuanjiezang@gmail.com)

1. **For an AJAX request, is there any difference between a GET and a POST HTTP request?**

Yes, there is a difference:

* **GET**: Retrieves data from the server. It sends data as part of the URL, which is visible and has a size limit.
* **POST**: Submits data to the server. The data is sent in the body of the HTTP request, which is not visible in the URL and has no size limitation.

1. **What does a REST service provide?**

A REST service provides a stateless interface for interacting with resources, typically over HTTP. It uses standard HTTP methods such as **GET**, **POST**, **PUT**, and **DELETE** to create, retrieve, update, or delete resources represented in formats like **JSON** or **XML**.

**3. How would you provide extra data with a request using jQuery's**

**$.get() method?**

pass additional data by providing it as the second argument in the $.get() method. Example:

$.get("example.com/api", { key: "value", id: 123 }, function(response) {

// Handle response

});

1. **How would you access a JSON object in a callback function provided to the $.getJSON() method?**

The JSON object is passed as a parameter to the callback function. You can access its properties directly. Example:

$.getJSON("example.com/data.json", function(data) {

console.log(data.propertyName); // Access JSON property

});

1. **Given the following XML, how would you access the contents of the <response> element, assuming you used jQuery's $.get() method to request the XML document?**

**<?xml version=*"1.0"* encoding=*"UTF-8"***

**standalone=*"yes"*?><response>Yes!</response>**

access the XML response using jQuery's DOM traversal methods. Example:

$.get("example.com/data.xml", function(data) {

var responseText = $(data).find("response").text();

console.log(responseText); // Outputs: Yes!

});

1. **If you wanted to load an HTML snippet into a selection of elements, what jQuery method would you use?**

access the XML response using jQuery's DOM traversal methods. Example:

$.get("example.com/data.xml", function(data) {

var responseText = $(data).find("response").text();

console.log(responseText); // Outputs: Yes!

});

**7. In the following JavaScript, describe the purpose of each property's callback function:$.ajaxSetup({ beforeSend : function() { }, success : function() { }, error : function() { }, complete :**

**function() { }});**

**$.ajaxSetup({**

**beforeSend: function() { },**

**success: function() { },**

**error: function() { },**

**complete: function() { }**

**});**

· **beforeSend**: This function is executed before the AJAX request is sent. It can be used to modify the request or display a loading indicator.

· **success**: This function is called if the AJAX request completes successfully. It processes the server's response.

· **error**: This function is triggered if the request fails. It handles error responses.

· **complete**: This function runs after the request completes, regardless of success or failure. It is often used for cleanup operations like hiding loading indicators.

**8. If you wanted to attach AJAX events to apply only in the context of an individual AJAX request, rather than globally, what methods does**

**jQuery provide to attach events in this way?**

use individual AJAX methods like .done(), .fail(), and .always() to attach events specific to a single AJAX request. Example:

$.ajax({

url: "example.com/api",

method: "GET"

}).done(function() {

// Success handler

}).fail(function() {

// Error handler

}).always(function() {

// Complete handler

});

1. **What jQuery method would you use if you wanted to get the value of every input element within a form?**

use .serialize() or iterate through the input elements using .each() to get the values. Example using .serialize():

var formData = $("form").serialize();console.log(formData); // Outputs a URL-encoded string of form data

**10.How would you go about implementing a client-side request to REST service providing the DELETE method where you pass a JSON object along to the server within the request? Describe the configuration you would need to accomplish this and then provide example code.**

To send a DELETE request with a JSON object, you can use the $.ajax() method. The configuration would include the type as DELETE and the data as the JSON object, typically in string format.

$.ajax({

url: "example.com/api/resource/123",

type: "DELETE",

data: JSON.stringify({ key: "value" }),

contentType: "application/json",

success: function(response) {

console.log("Resource deleted:", response);

},

error: function(error) {

console.log("Error:", error);

}

});