Chapter 12

How to use Ajax, JSON, and Flickr

Objectives

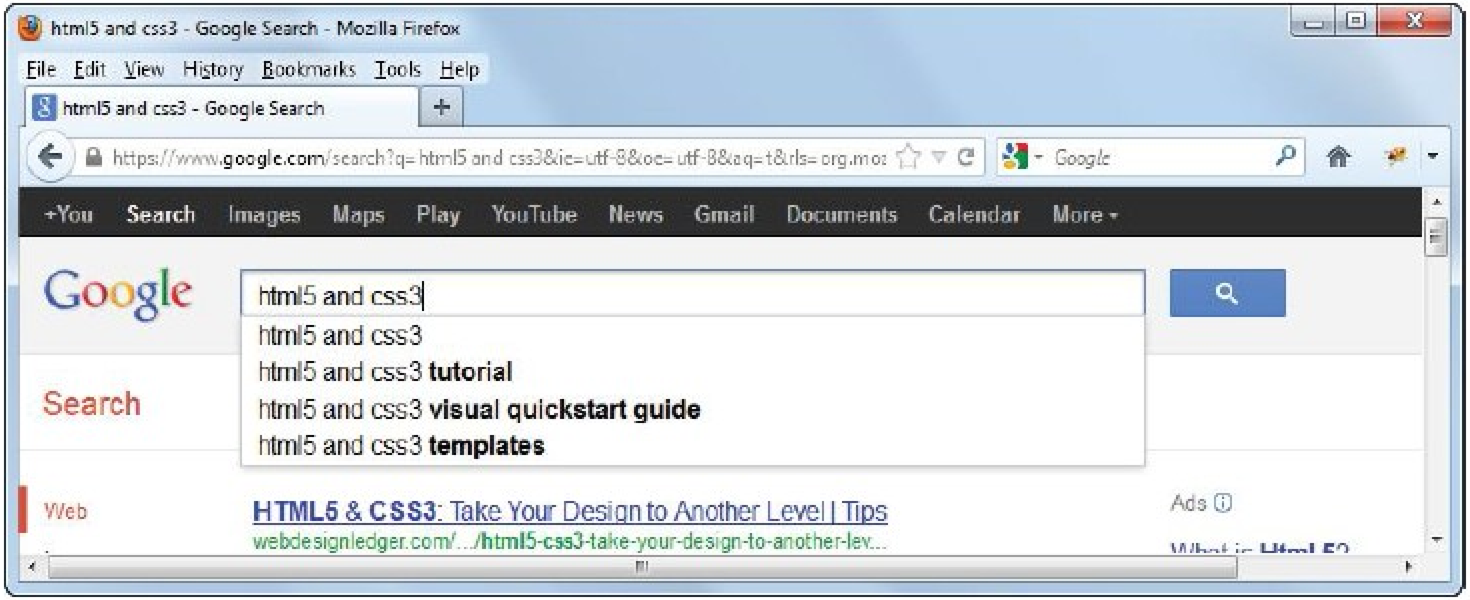
- Introduction to Ajax
- How to use the jQuery shorthand methods for Ajax
- How to use the $.ajax() method for working with Ajax
- How to use Ajax with Flickr

Introduction to Ajax

Introduction to Ajax

- ***Ajax*** stands for *Asynchronous JavaScript and XML*.
- Unlike normal HTTP request, Ajax let you receive data from web server without reloading the page.

Google's Auto Suggest is an Ajax application

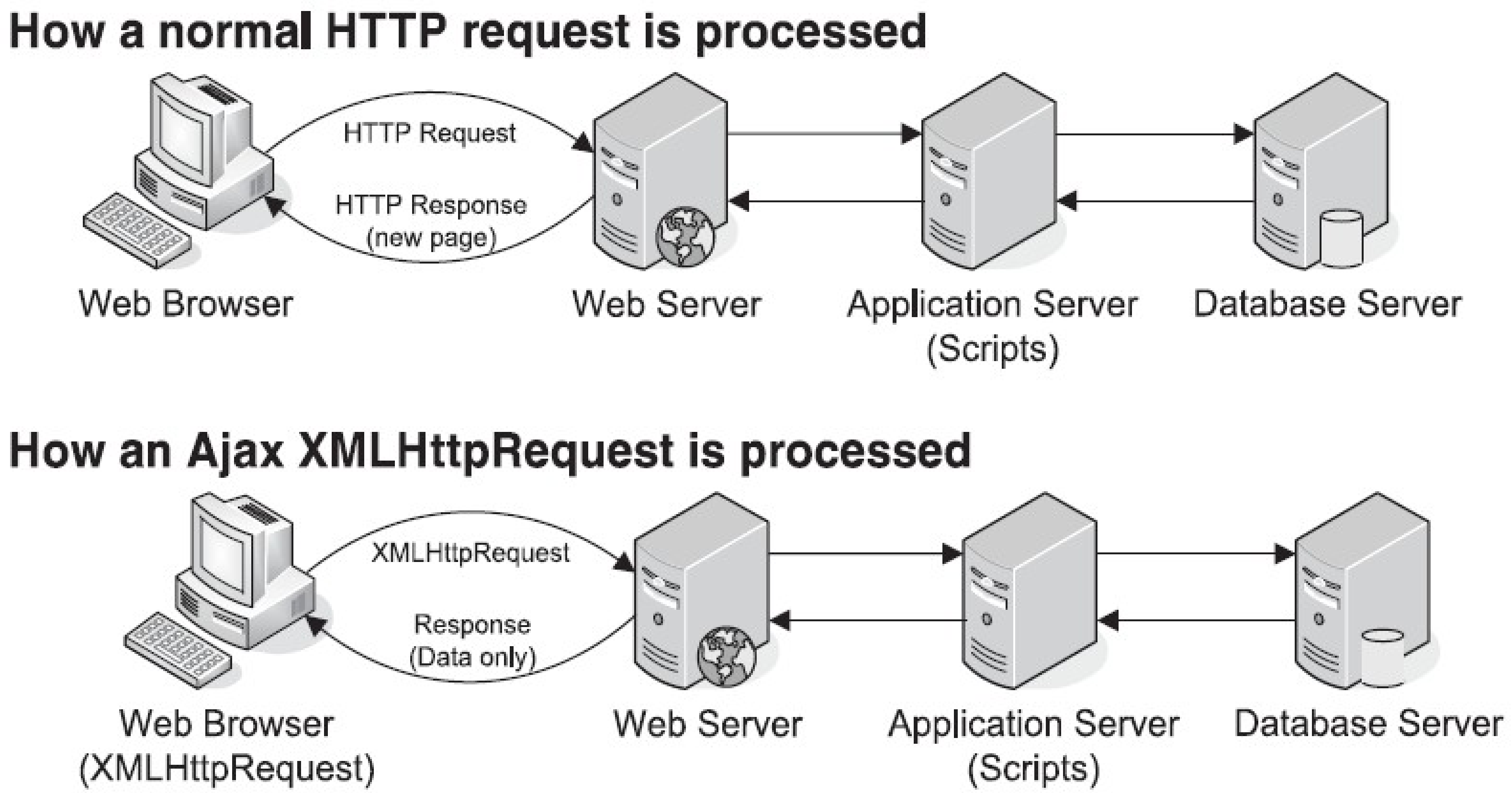

html5 and css3 - Google Search - Mozilla Firefox
File Edit View History Bookmarks Tools Help
html5 and css3 - Google Search
+
https://www.google.com/search?q=html5 and css3&ie=utf-8&oe=utf-8&aq=t&rls=org.moz
Google
+You Search Images Maps Play YouTube News Gmail Documents Calendar More
Google
Search
Web
Ads

html5 and css3

html5 and css3
html5 and css3 **tutorial**
html5 and css3 **visual quickstart guide**
html5 and css3 **templates**

HTML5 & CSS3: Take Your Design to Another Level | Tips
webdesignledger.com/.../html5-css3-take-your-design-to-another-lev...

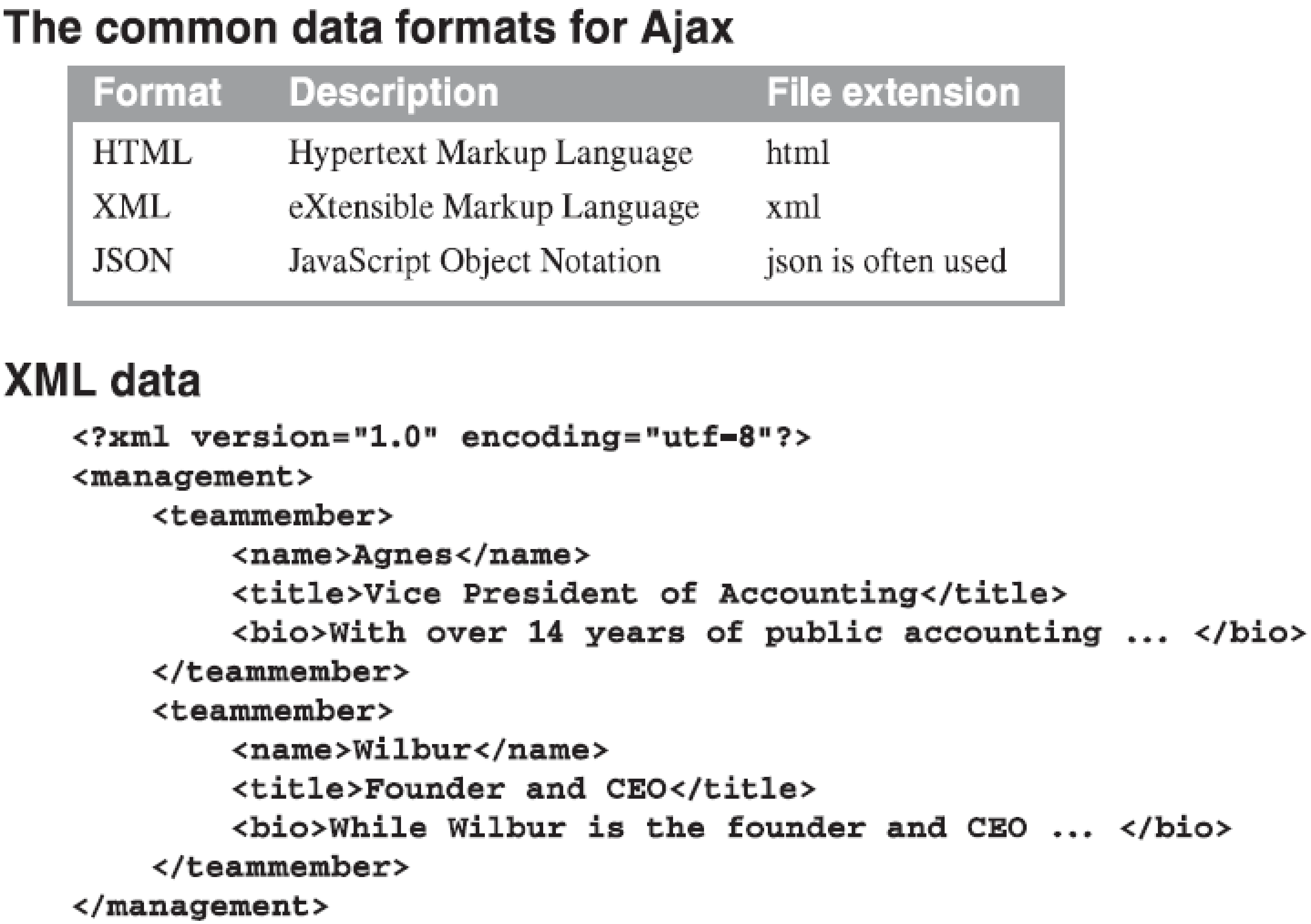
How Ajax works



How Ajax works (cont.)

- Ajax need to use JavaScript to send the request, process the responses and updates the DOM with the new data.
- To send an Ajax request, JavaScript uses a browser object known as XMLHttpRequest (XHR) object.
- XHR Object can include data that tells the application server what data is being requested.

Common data formats for Ajax



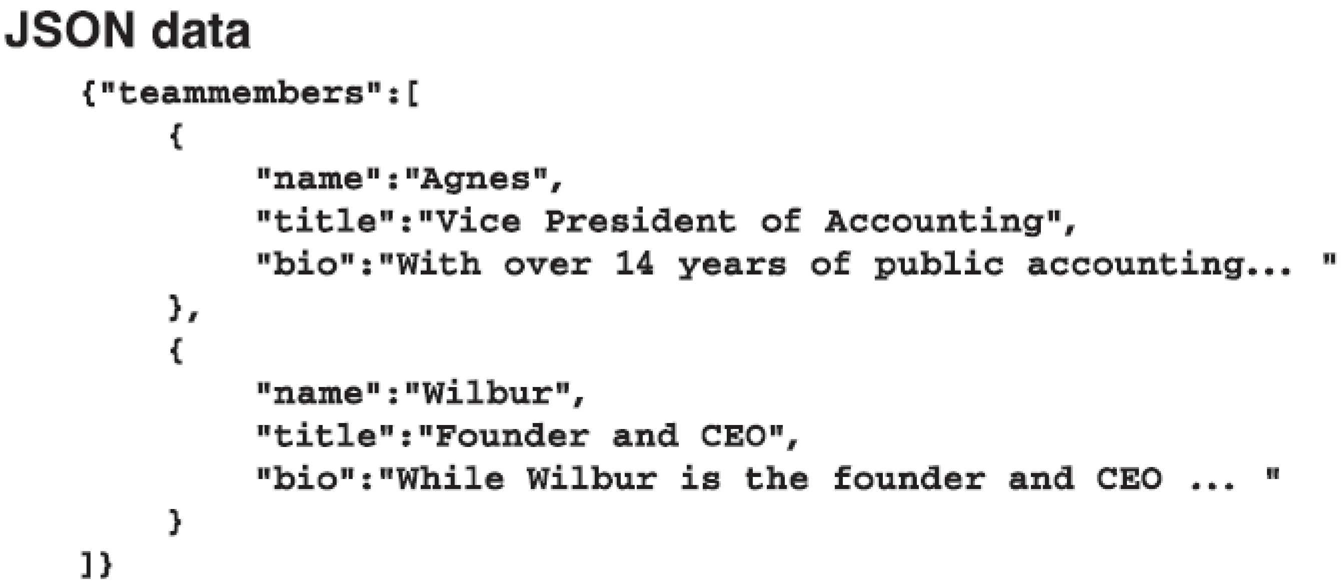
The common data formats for Ajax

Format	Description	File extension
HTML	Hypertext Markup Language	html
XML	eXtensible Markup Language	xml
JSON	JavaScript Object Notation	json is often used

XML data

```
<?xml version="1.0" encoding="utf-8"?>
<management>
    <teammember>
        <name>Agnes</name>
        <title>Vice President of Accounting</title>
        <bio>With over 14 years of public accounting ... </bio>
    </teammember>
    <teammember>
        <name>Wilbur</name>
        <title>Founder and CEO</title>
        <bio>While Wilbur is the founder and CEO ... </bio>
    </teammember>
</management>
```

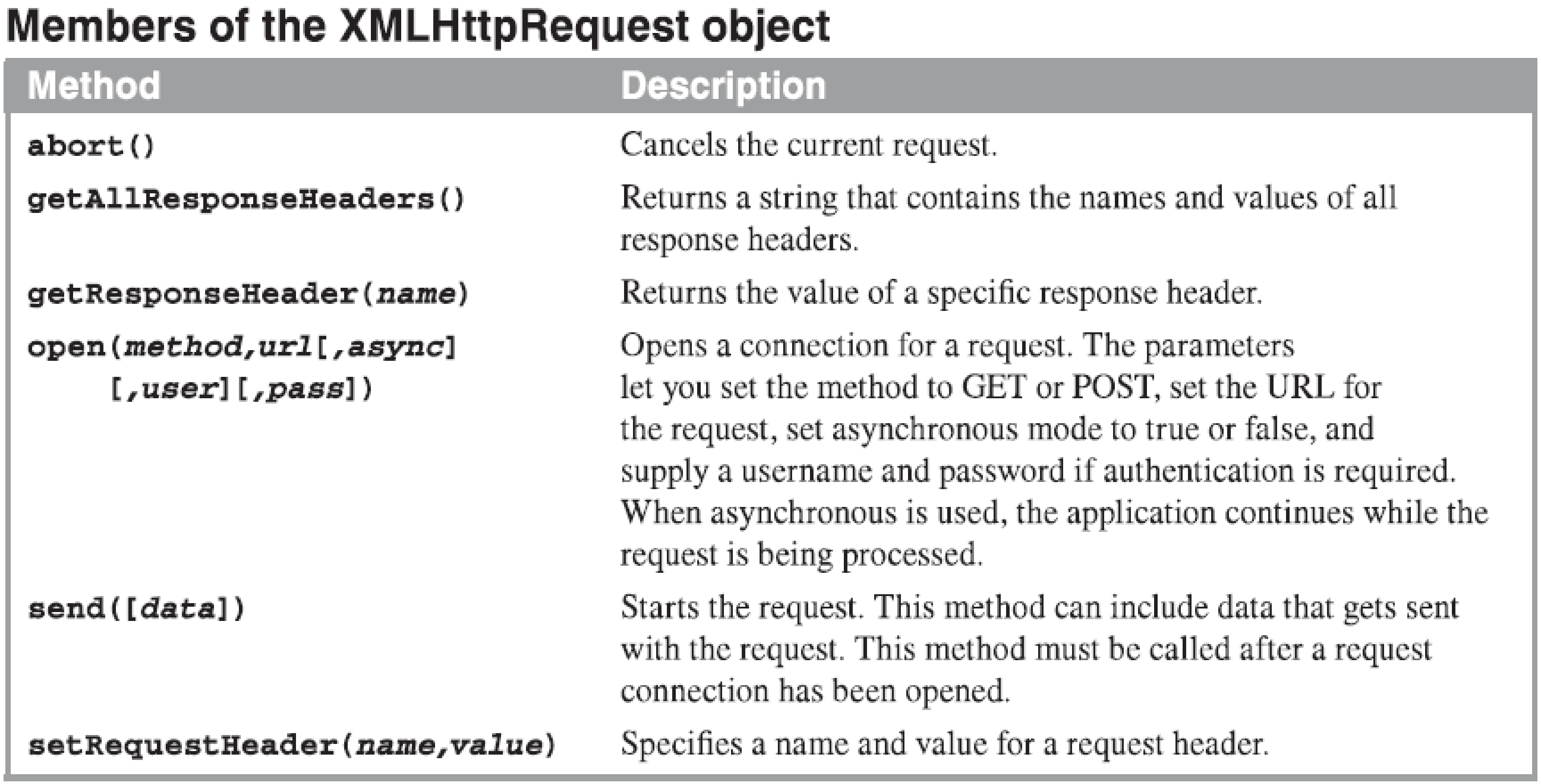
Common data formats for Ajax (cont.)



JSON data

```
{"teammembers":[
     {
          "name":"Agnes",
          "title":"Vice President of Accounting",
          "bio":"With over 14 years of public accounting... "
     },
     {
          "name":"Wilbur",
          "title":"Founder and CEO",
          "bio":"While Wilbur is the founder and CEO ... "
     }
]}
```

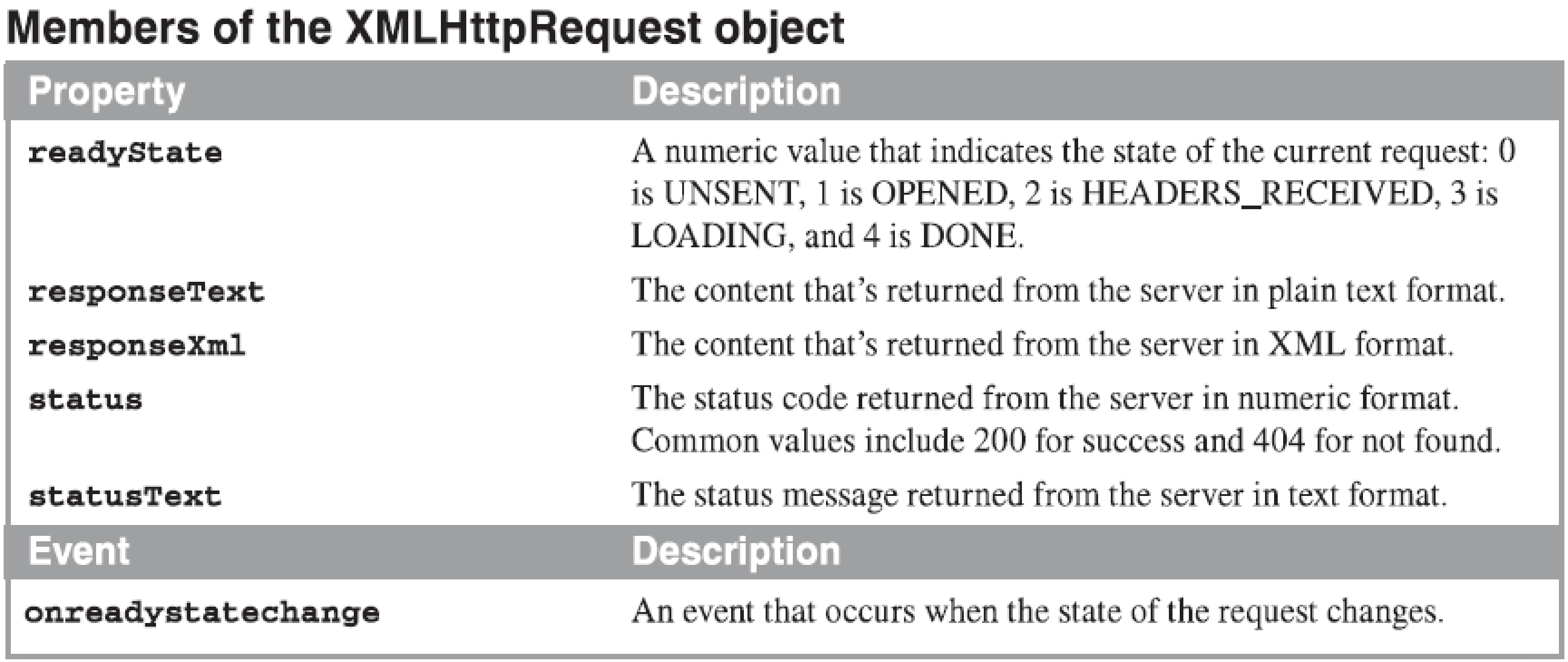
The members of the XMLHttpRequest object



Members of the XMLHttpRequest object

Method	Description
abort()	Cancels the current request.
getAllResponseHeaders()	Returns a string that contains the names and values of all response headers.
getResponseHeader(*name*)	Returns the value of a specific response header.
open(*method,url*[*,async*] [*,user*][*,pass*])	Opens a connection for a request. The parameters let you set the method to GET or POST, set the URL for the request, set asynchronous mode to true or false, and supply a username and password if authentication is required. When asynchronous is used, the application continues while the request is being processed.
send([*data*])	Starts the request. This method can include data that gets sent with the request. This method must be called after a request connection has been opened.
setRequestHeader(*name,value*)	Specifies a name and value for a request header.

The members of the XMLHttpRequest object (cont.)



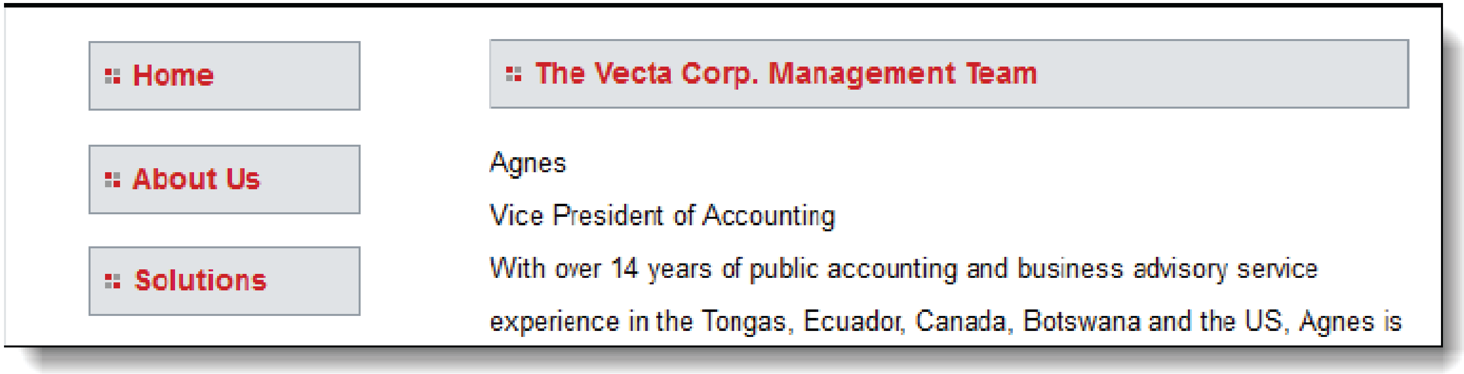
Members of the XMLHttpRequest object

Property	Description
`readyState`	A numeric value that indicates the state of the current request: 0 is UNSENT, 1 is OPENED, 2 is HEADERS_RECEIVED, 3 is LOADING, and 4 is DONE.
`responseText`	The content that's returned from the server in plain text format.
`responseXml`	The content that's returned from the server in XML format.
`status`	The status code returned from the server in numeric format. Common values include 200 for success and 404 for not found.
`statusText`	The status message returned from the server in text format.
Event	**Description**
`onreadystatechange`	An event that occurs when the state of the request changes.

How to use the XMLHttpRequest object

- This application uses the XHR object to load all of the team members in the file named team.xml on the web server and display them.

A web page that uses the XHR object and JavaScript to load XML data

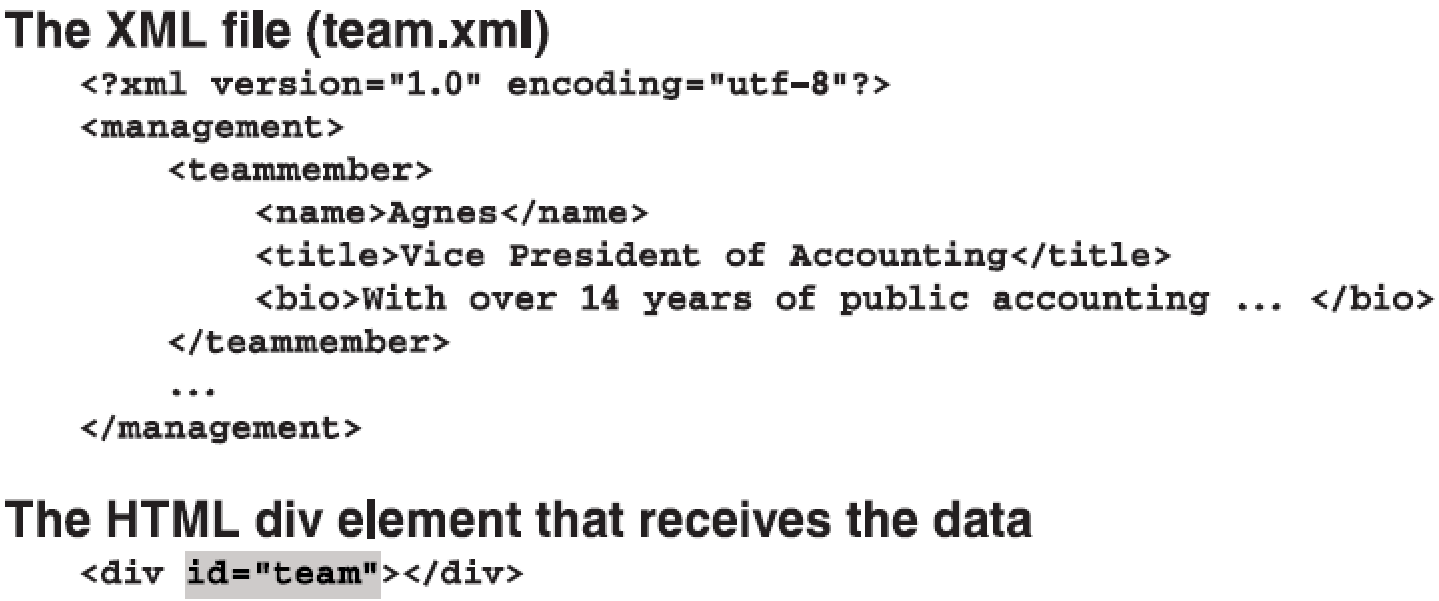


Agnes

Vice President of Accounting

With over 14 years of public accounting and business advisory service experience in the Tongas, Ecuador, Canada, Botswana and the US, Agnes is

How to use the XMLHttpRequest object (cont.)



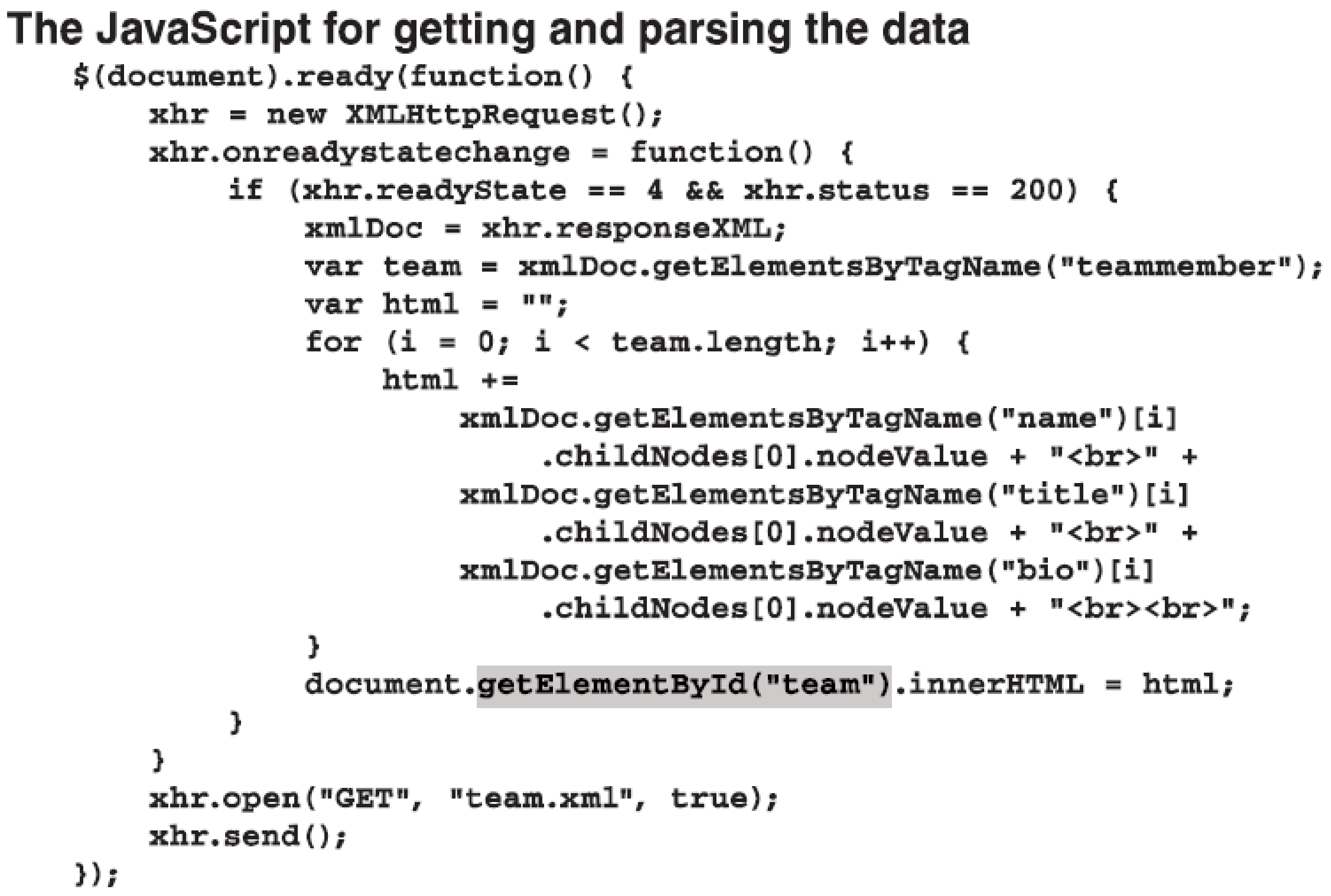
The XML file (team.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<management>
    <teammember>
        <name>Agnes</name>
        <title>Vice President of Accounting</title>
        <bio>With over 14 years of public accounting ... </bio>
    </teammember>
    ...
</management>
```

The HTML div element that receives the data

```
<div id="team"></div>
```

How to use the XMLHttpRequest object (cont.)



The JavaScript for getting and parsing the data

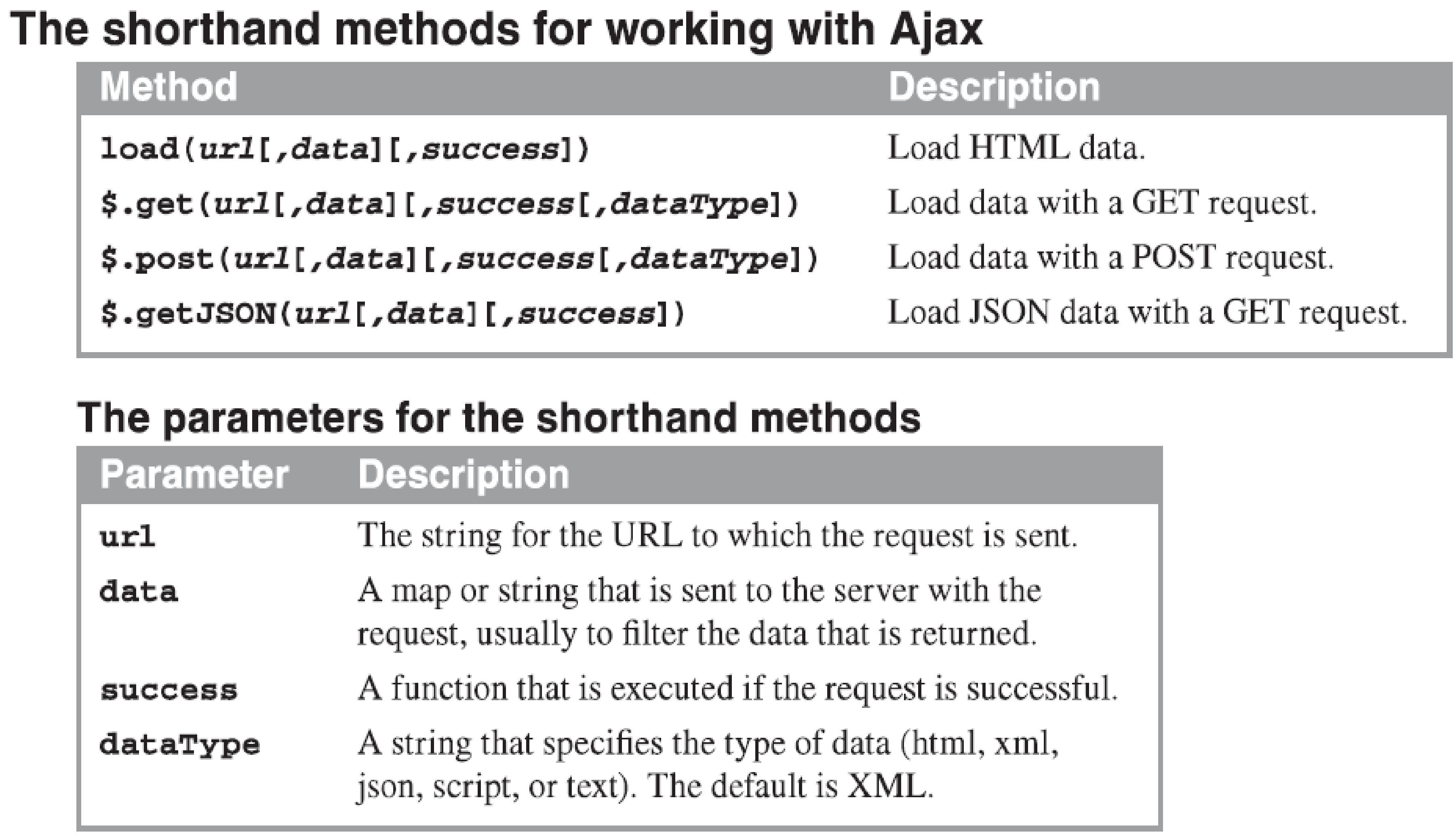
```
$(document).ready(function() {
    xhr = new XMLHttpRequest();
    xhr.onreadystatechange = function() {
        if (xhr.readyState == 4 && xhr.status == 200) {
            xmlDoc = xhr.responseXML;
            var team = xmlDoc.getElementsByTagName("teammember");
            var html = "";
            for (i = 0; i < team.length; i++) {
                html +=
                    xmlDoc.getElementsByTagName("name")[i]
                        .childNodes[0].nodeValue + "<br>" +
                    xmlDoc.getElementsByTagName("title")[i]
                        .childNodes[0].nodeValue + "<br>" +
                    xmlDoc.getElementsByTagName("bio")[i]
                        .childNodes[0].nodeValue + "<br><br>";
            }
            document.getElementById("team").innerHTML = html;
        }
    }
    xhr.open("GET", "team.xml", true);
    xhr.send();
});
```

How to use the jQuery shorthand methods for Ajax

The jQuery shorthand methods for working with Ajax

- jQuery includes several shorthand methods that let you request and receive HTML, XML or JSON data.
- All of the shorthand methods let you include data that will be used by the web server to filter the results of the request so only right the results are returned.
- The $.get() and $.post() method are used for request (GET or POST).
- The $each() method is an expanded form of the each() method.

The jQuery shorthand methods for working with Ajax (cont.)



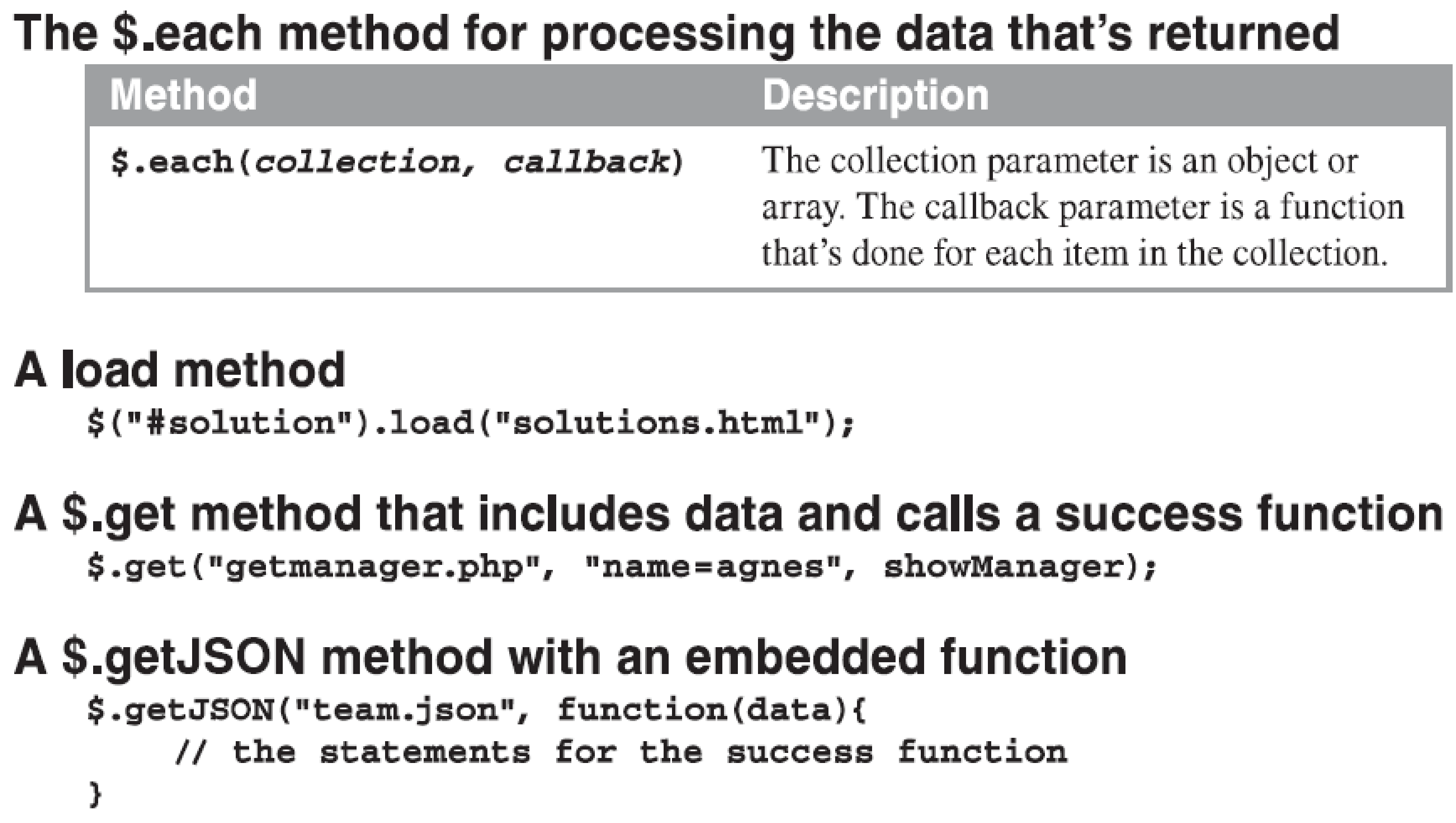
The shorthand methods for working with Ajax

Method	Description
`load(url[,data][,success])`	Load HTML data.
`$.get(url[,data][,success[,dataType])`	Load data with a GET request.
`$.post(url[,data][,success[,dataType])`	Load data with a POST request.
`$.getJSON(url[,data][,success])`	Load JSON data with a GET request.

The parameters for the shorthand methods

Parameter	Description
`url`	The string for the URL to which the request is sent.
`data`	A map or string that is sent to the server with the request, usually to filter the data that is returned.
`success`	A function that is executed if the request is successful.
`dataType`	A string that specifies the type of data (html, xml, json, script, or text). The default is XML.

The jQuery shorthand methods for working with Ajax (cont.)



The $.each method for processing the data that's returned

Method	Description
`$.each(collection, callback)`	The collection parameter is an object or array. The callback parameter is a function that's done for each item in the collection.

A load method

```
$("#solution").load("solutions.html");
```

A $.get method that includes data and calls a success function

```
$.get("getmanager.php", "name=agnes", showManager);
```

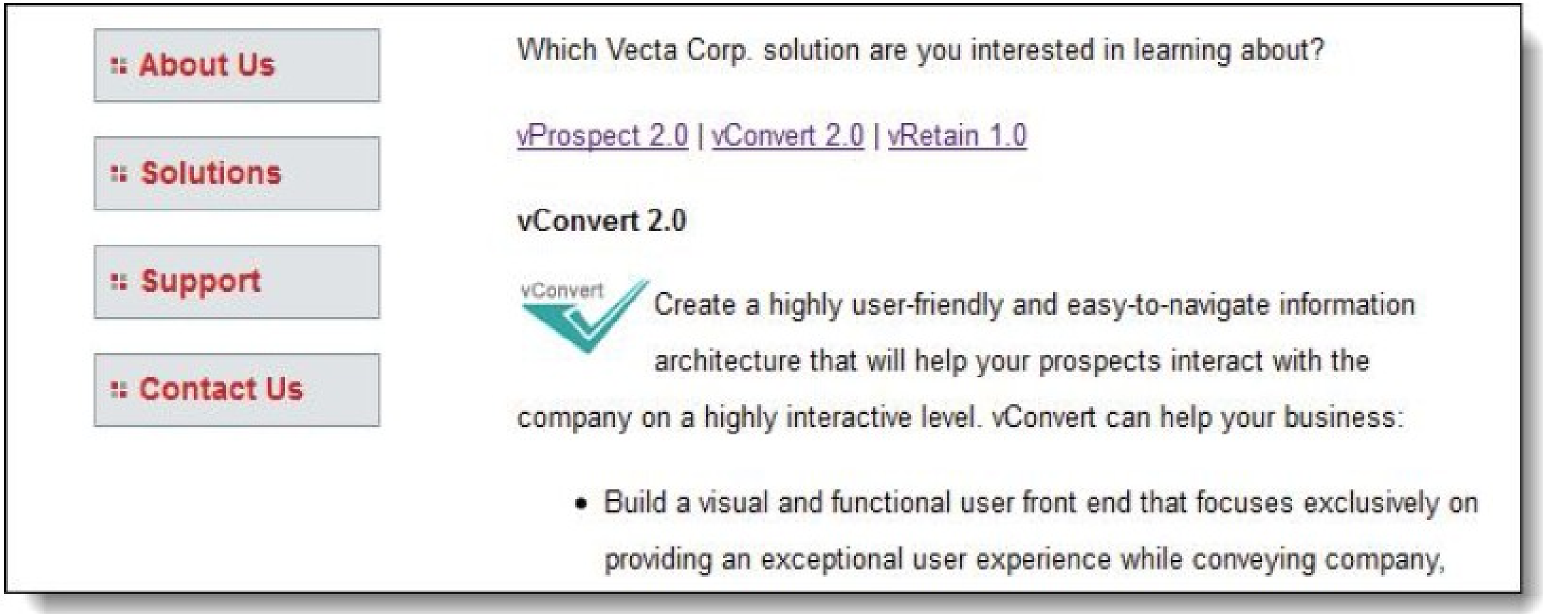
A $.getJSON method with an embedded function

```
$.getJSON("team.json", function(data){
    // the statements for the success function
}
```

How to use the load() method to load HTML data

- The load() function can only load content from files on the same server as the page making the call.

A web page that loads HTML elements when one of the links is clicked



Which Vecta Corp. solution are you interested in learning about?

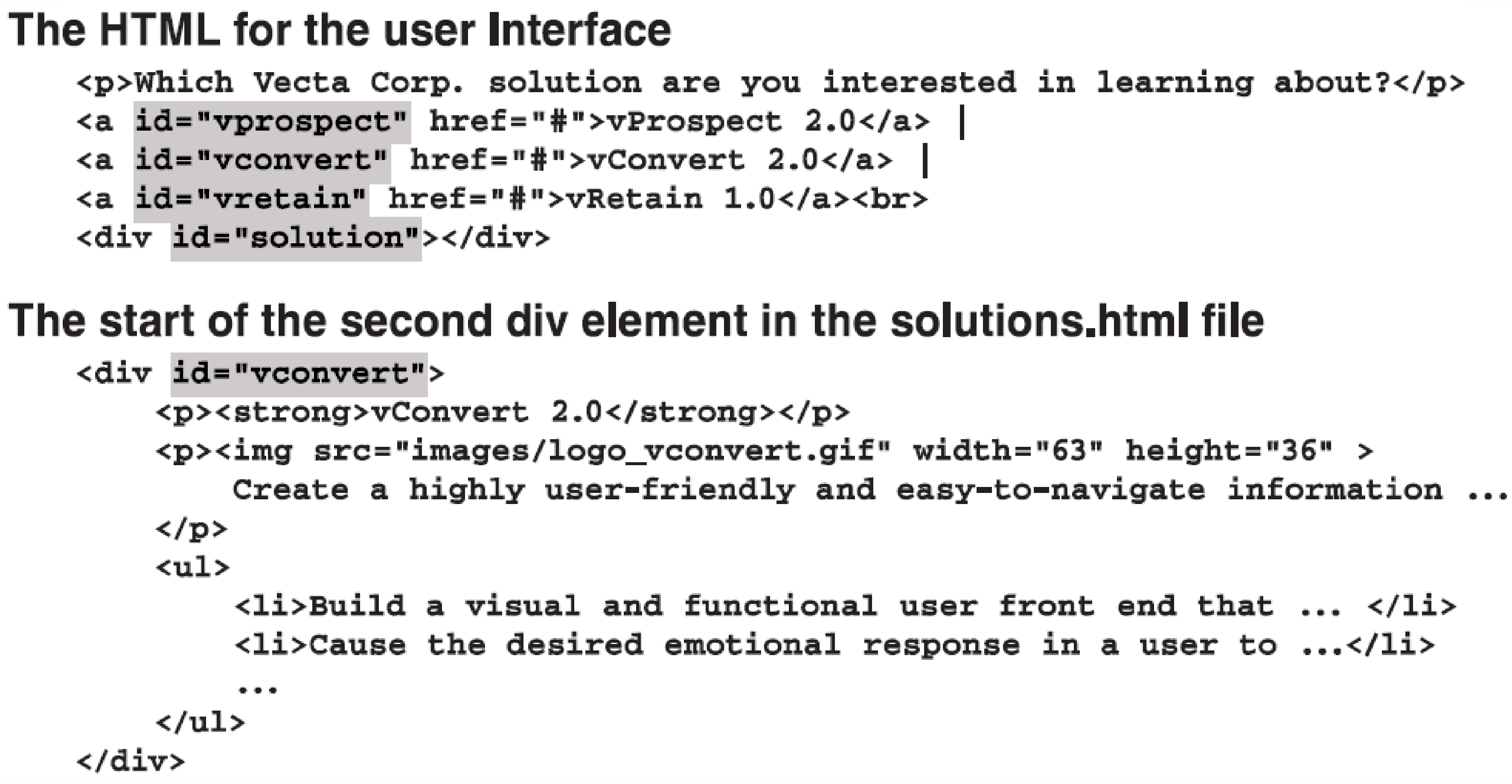
vProspect 2.0 | vConvert 2.0 | vRetain 1.0

vConvert 2.0

Create a highly user-friendly and easy-to-navigate information architecture that will help your prospects interact with the company on a highly interactive level. vConvert can help your business:

- Build a visual and functional user front end that focuses exclusively on providing an exceptional user experience while conveying company,

How to use the load() method to load HTML data (cont.)



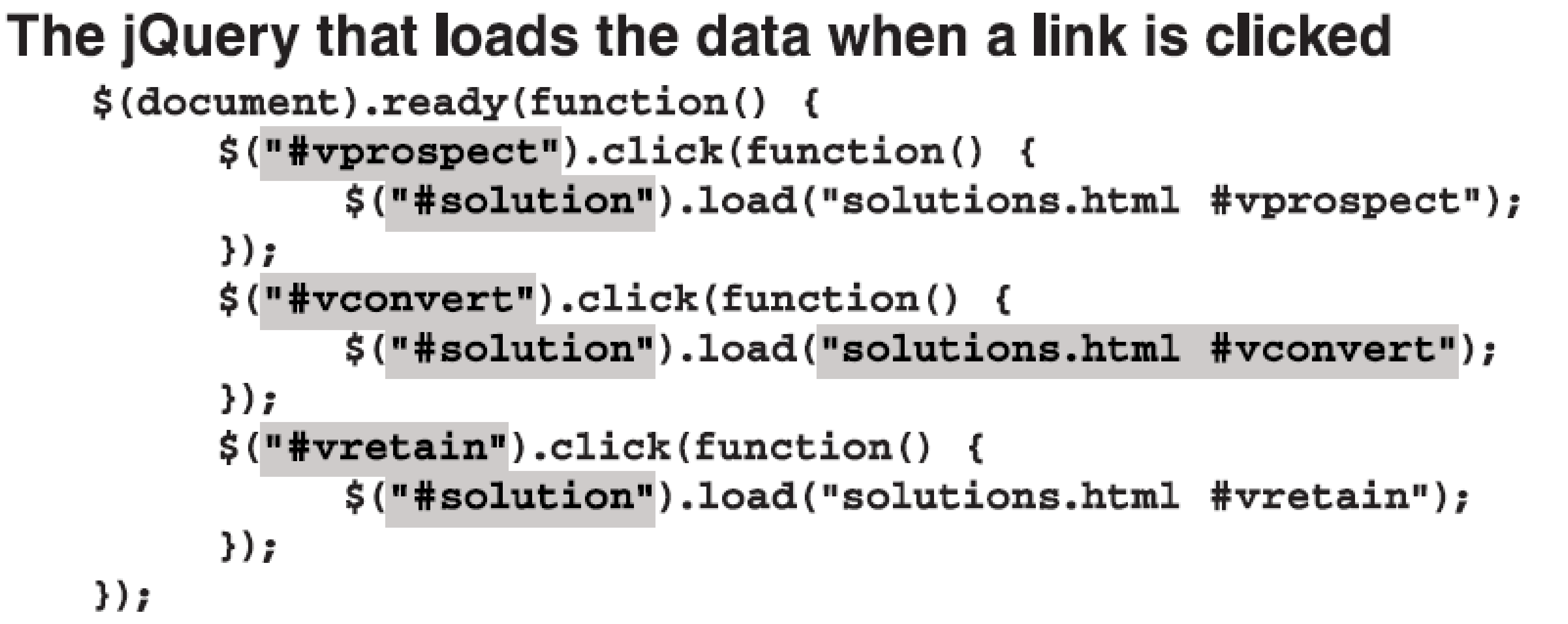
The HTML for the user Interface

```
<p>Which Vecta Corp. solution are you interested in learning about?</p>
<a id="vprospect" href="#">vProspect 2.0</a> |
<a id="vconvert" href="#">vConvert 2.0</a> |
<a id="vretain" href="#">vRetain 1.0</a><br>
<div id="solution"></div>
```

The start of the second div element in the solutions.html file

```
<div id="vconvert">
    <p><strong>vConvert 2.0</strong></p>
    <p><img src="images/logo_vconvert.gif" width="63" height="36" >
        Create a highly user-friendly and easy-to-navigate information ...
    </p>
    <ul>
        <li>Build a visual and functional user front end that ... </li>
        <li>Cause the desired emotional response in a user to ...</li>
        ...
    </ul>
</div>
```

How to use the load() method to load HTML data (cont.)

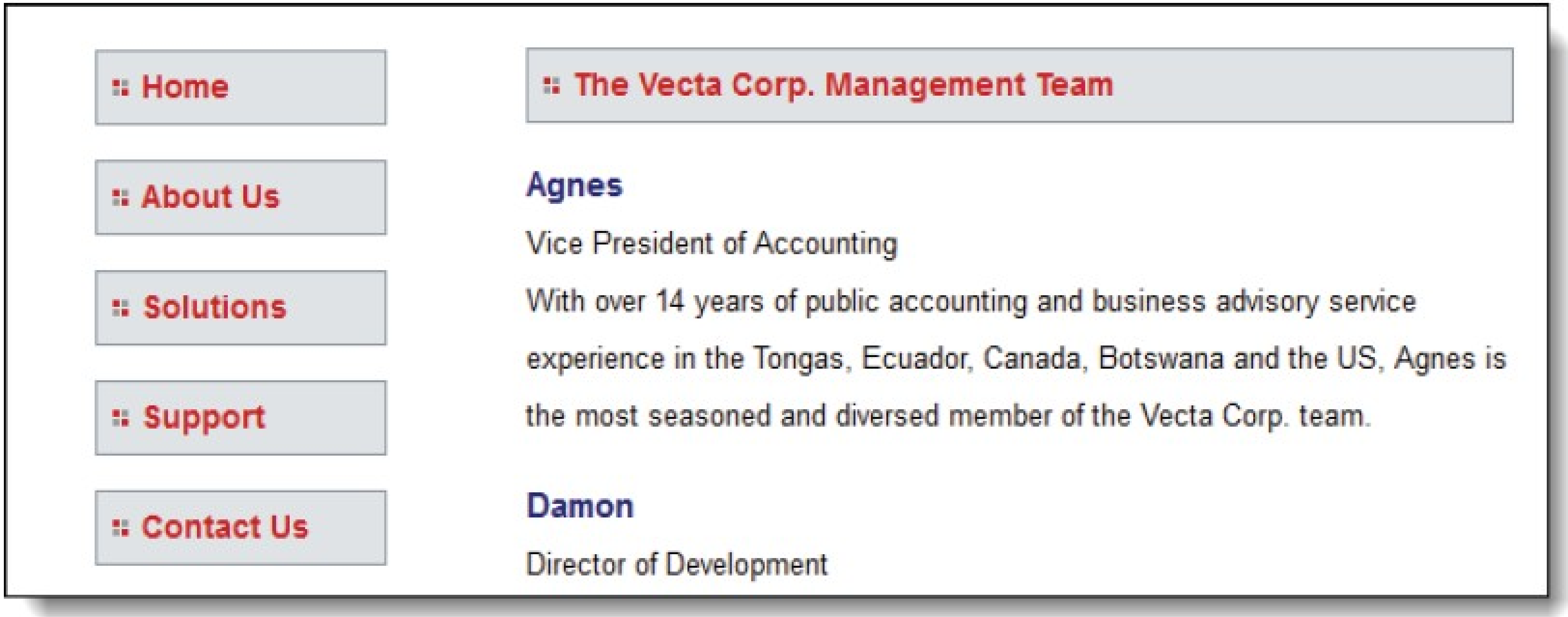


The jQuery that loads the data when a link is clicked

```
$(document).ready(function() {
    $("#vprospect").click(function() {
        $("#solution").load("solutions.html #vprospect");
    });
    $("#vconvert").click(function() {
        $("#solution").load("solutions.html #vconvert");
    });
    $("#vretain").click(function() {
        $("#solution").load("solutions.html #vretain");
    });
});
```

How to use the $.get() or $.post() method to load XML data

A web page that loads XML data



:: The Vecta Corp. Management Team

Agnes

Vice President of Accounting

With over 14 years of public accounting and business advisory service experience in the Tongas, Ecuador, Canada, Botswana and the US, Agnes is the most seasoned and diversed member of the Vecta Corp. team.

Damon

Director of Development

The XML file (team.xml)

```
<management>
    <teammember>
        <name>Agnes</name>
        <title>Vice President of Accounting</title>
        <bio>With over 14 years of public accounting ... </bio>
    </teammember>
    ...
</management>
```

The HTML div element that receives the data

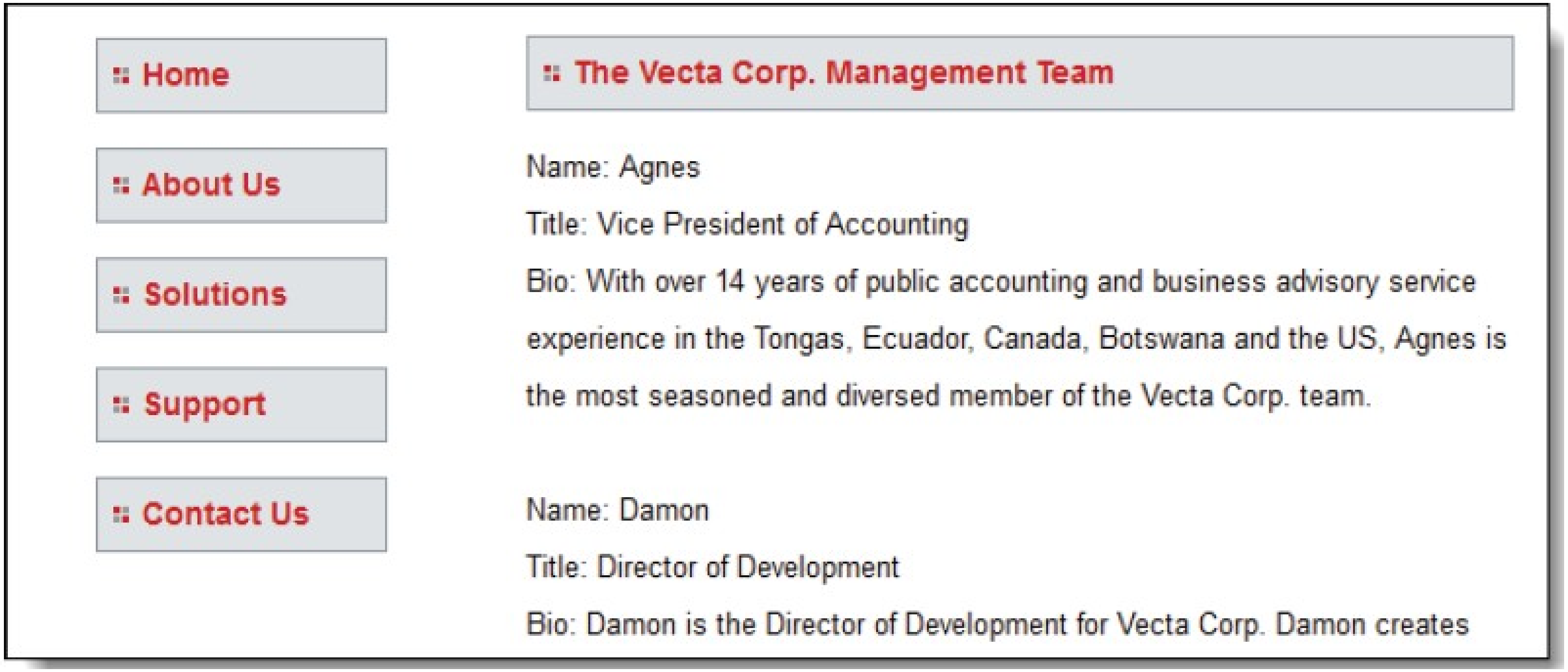
```
<div id="team"></div>
```

The jQuery

```
$(document).ready(function(){
    $.get("team.xml", function(data){
        $("#team").html("");
        $(data).find("management").children().each(function() {
            var xmlDoc = $(this);
            $("#team").append("<h3>" +
                xmlDoc.find("name").text() + "</h3>" +
                xmlDoc.find("title").text() + "<br>" +
                xmlDoc.find("bio").text() + "<br>");
        });
    });
});
```

How to use the $.getJSON() method to load JSON data

A web page that loads JSON data



:: The Vecta Corp. Management Team

Name: Agnes

Title: Vice President of Accounting

Bio: With over 14 years of public accounting and business advisory service experience in the Tongas, Ecuador, Canada, Botswana and the US, Agnes is the most seasoned and diversed member of the Vecta Corp. team.

Name: Damon

Title: Director of Development

Bio: Damon is the Director of Development for Vecta Corp. Damon creates

The JSON file (team.json)

```
{"teammembers":[
    {
        "name":"Agnes",
        "title":"Vice President of Accounting",
        "bio":"With over 14 years of public accounting... "
    },
    {
        "name":"Damon",
        "title":"Director of Development",
        "bio":"Damon is the Director of Development for ... "
    }
]}
```

The HTML div element that receives the data

```
<div id="team"></div>
```

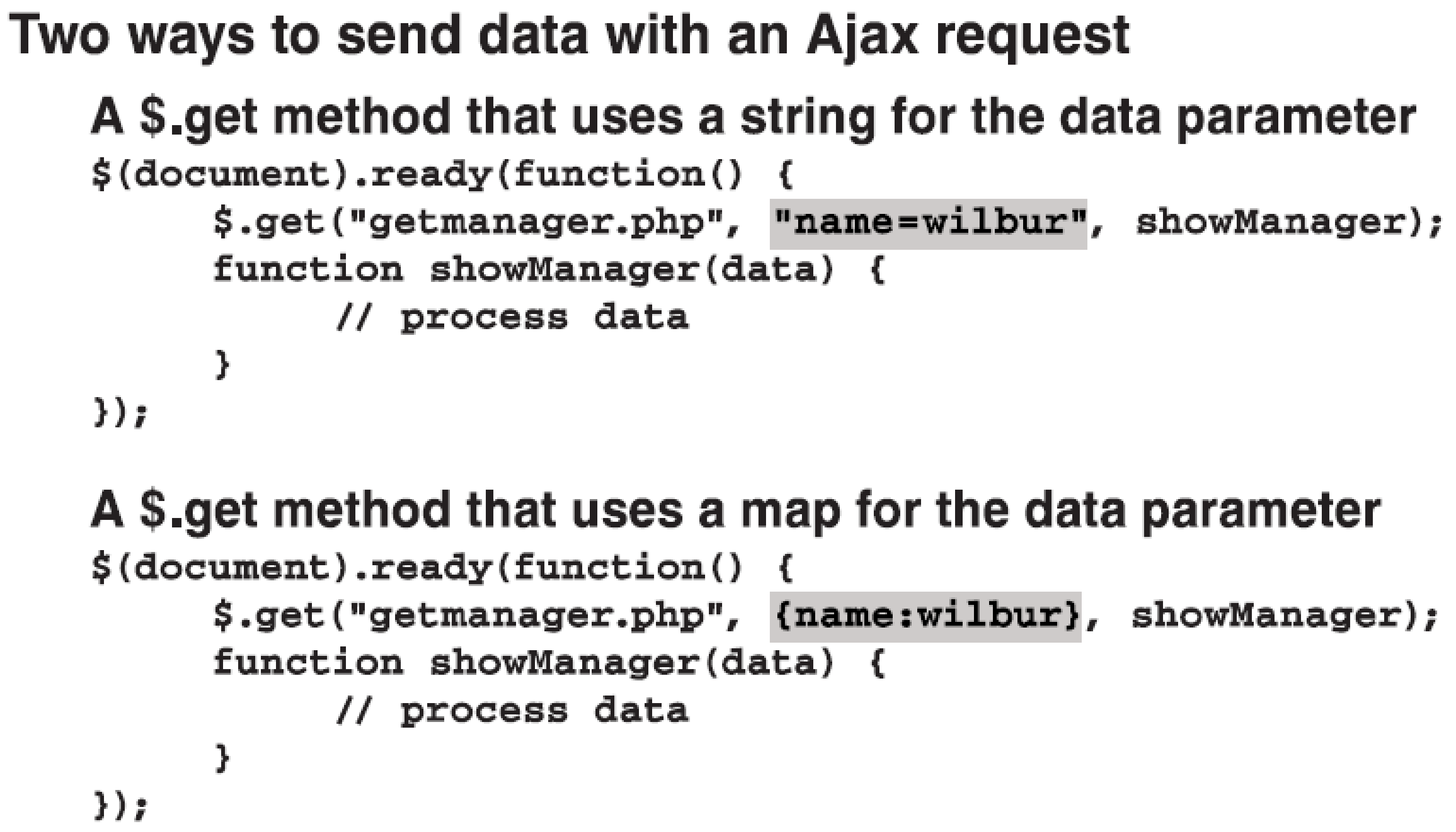
The jQuery

```
$(document).ready(function(){
    $.getJSON("team.json", function(data){
        $.each(data, function() {
            $.each(this, function(key, value) {
                $("#team").append(
                    "Name: " + value.name + "<br>" +
                    "Title: " + value.title + "<br>" +
                    "Bio: " + value.bio + "<br><br>"
                );
            });
        });
    });
});
```

How to send data with an Ajax request

- When you send data with an Ajax request, the URL is for a server-side script such as a PHP file. Then, the script is responsible for returning the data in XML or JSON format.
- The data parameter in a jQuery shortcut method is a name/value pair that can be set either as query string or a map.
- The jQuery helper functions for Ajax make it easy to package form data before sending it to the server.

How to send data with an Ajax request (cont.)



Two ways to send data with an Ajax request

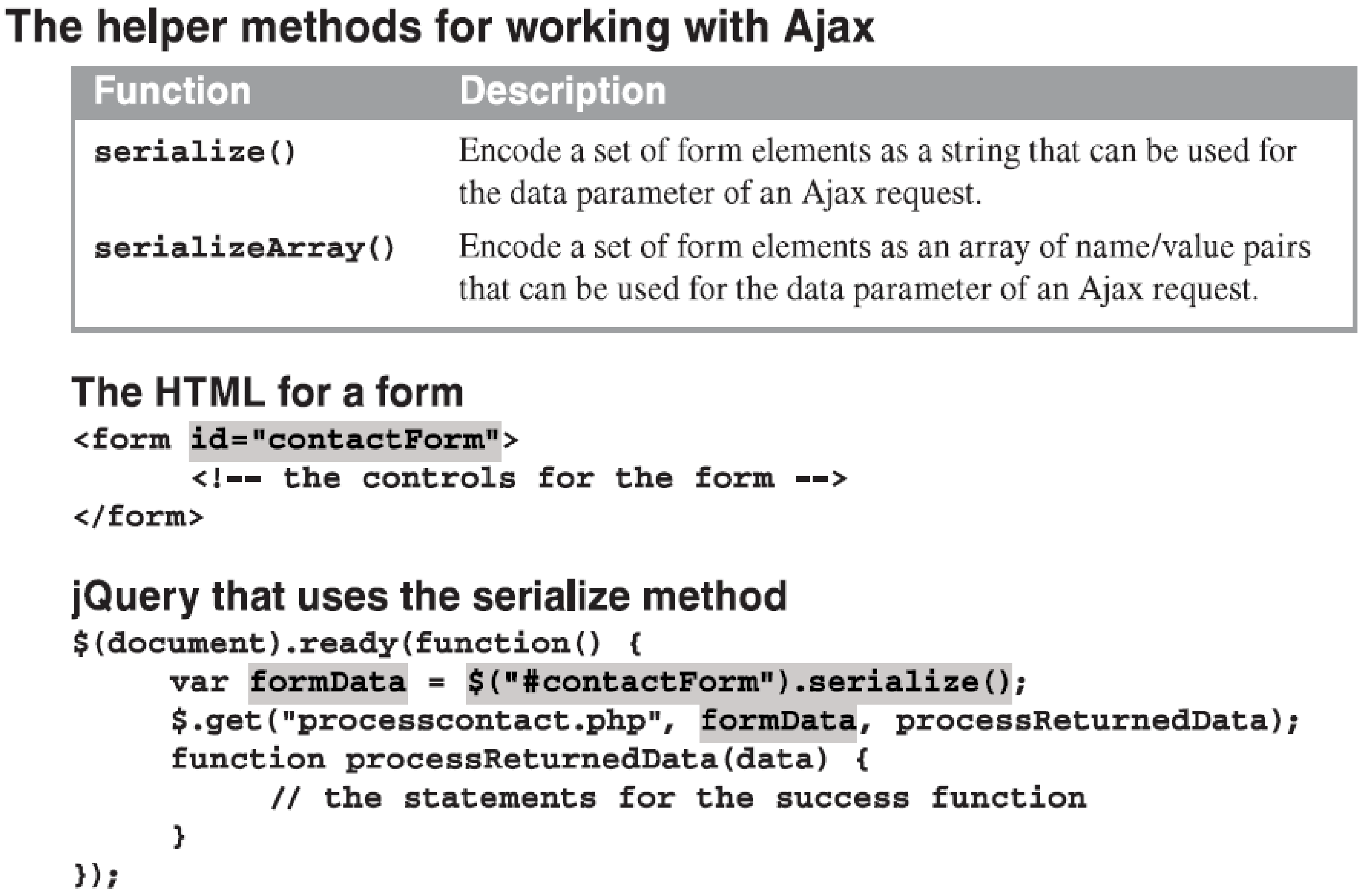
A $.get method that uses a string for the data parameter

```
$(document).ready(function() {
      $.get("getmanager.php", "name=wilbur", showManager);
      function showManager(data) {
            // process data
      }
});
```

A $.get method that uses a map for the data parameter

```
$(document).ready(function() {
      $.get("getmanager.php", {name:wilbur}, showManager);
      function showManager(data) {
            // process data
      }
});
```

How to send data with an Ajax request (cont.)



The helper methods for working with Ajax

Function	Description
`serialize()`	Encode a set of form elements as a string that can be used for the data parameter of an Ajax request.
`serializeArray()`	Encode a set of form elements as an array of name/value pairs that can be used for the data parameter of an Ajax request.

The HTML for a form

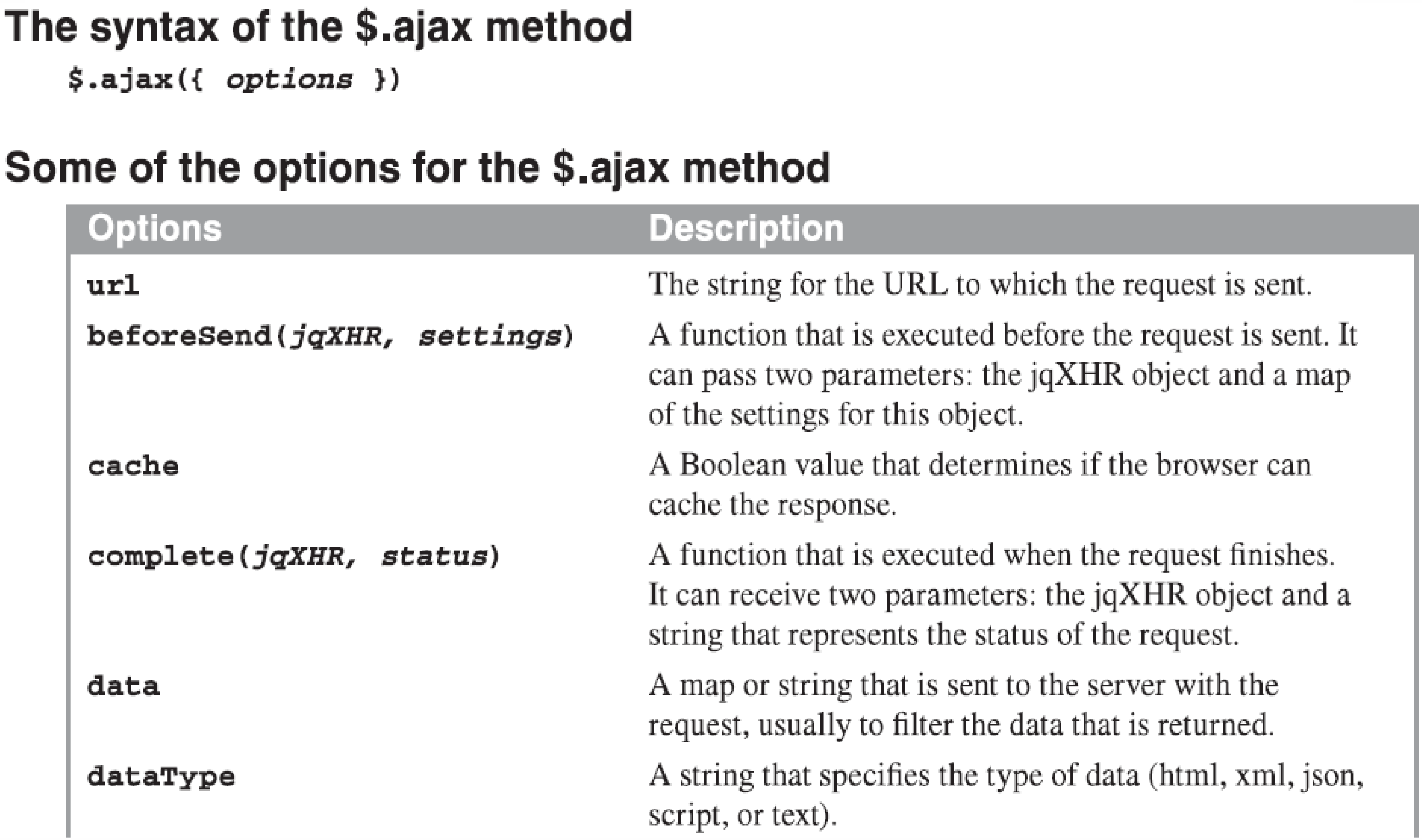
```
<form id="contactForm">
      <!-- the controls for the form -->
</form>
```

jQuery that uses the serialize method

```
$(document).ready(function() {
     var formData = $("#contactForm").serialize();
     $.get("processcontact.php", formData, processReturnedData);
     function processReturnedData(data) {
          // the statements for the success function
     }
});
```

How to use the $.ajax() method for working with Ajax

The syntax of the $.ajax() method



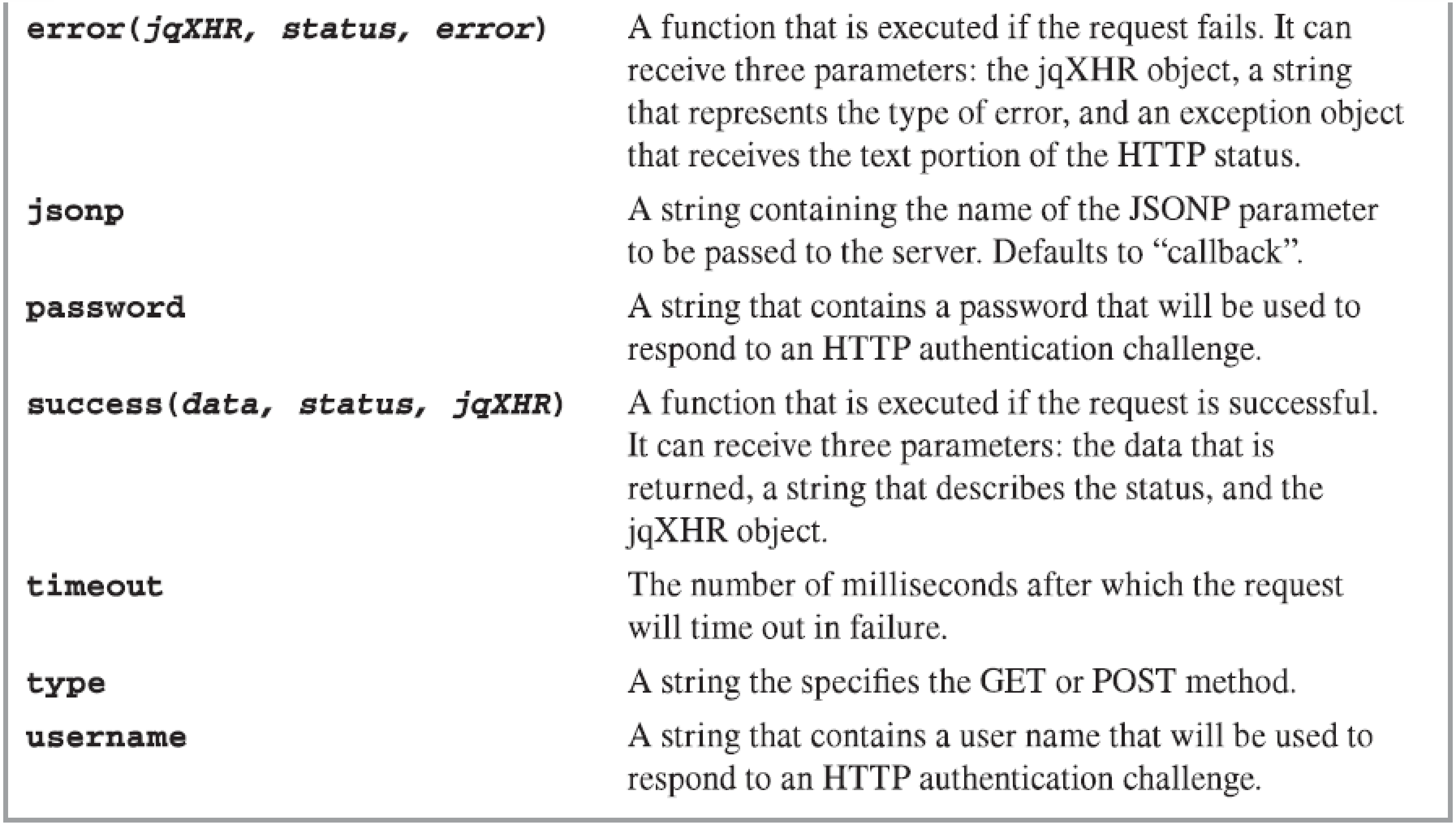
The syntax of the $.ajax method

```
$.ajax({ options })
```

Some of the options for the $.ajax method

Options	Description
`url`	The string for the URL to which the request is sent.
`beforeSend(jqXHR, settings)`	A function that is executed before the request is sent. It can pass two parameters: the jqXHR object and a map of the settings for this object.
`cache`	A Boolean value that determines if the browser can cache the response.
`complete(jqXHR, status)`	A function that is executed when the request finishes. It can receive two parameters: the jqXHR object and a string that represents the status of the request.
`data`	A map or string that is sent to the server with the request, usually to filter the data that is returned.
`dataType`	A string that specifies the type of data (html, xml, json, script, or text).

The syntax of the $.ajax() method (cont.)



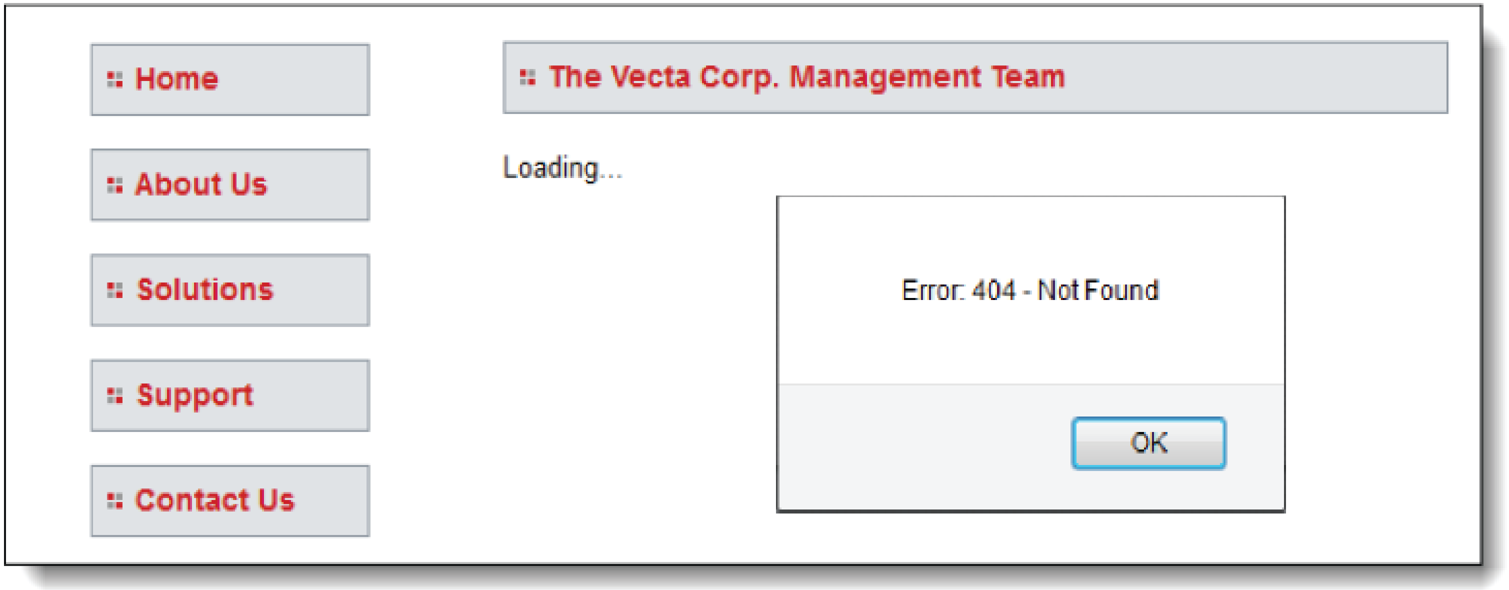
`error(jqXHR, status, error)`	A function that is executed if the request fails. It can receive three parameters: the jqXHR object, a string that represents the type of error, and an exception object that receives the text portion of the HTTP status.
`jsonp`	A string containing the name of the JSONP parameter to be passed to the server. Defaults to "callback".
`password`	A string that contains a password that will be used to respond to an HTTP authentication challenge.
`success(data, status, jqXHR)`	A function that is executed if the request is successful. It can receive three parameters: the data that is returned, a string that describes the status, and the jqXHR object.
`timeout`	The number of milliseconds after which the request will time out in failure.
`type`	A string the specifies the GET or POST method.
`username`	A string that contains a user name that will be used to respond to an HTTP authentication challenge.

The syntax of the $.ajax() method (cont.)

- The **$.ajax()** method provides options that give you more control over the way the Ajax request works, such as providing a function for handling errors.
- The **jqXHR** object is jQuery’s superset of the standard XMLHttpRequest object that provide properties of that project.
- JSONP or “JSON with padding” is a complement to JSON data that let you request data from a server in a different domain.

How to use the $.ajax() method to load data

A web page with a loading message and an alert dialog box for errors



The Vecta Corp. Management Team

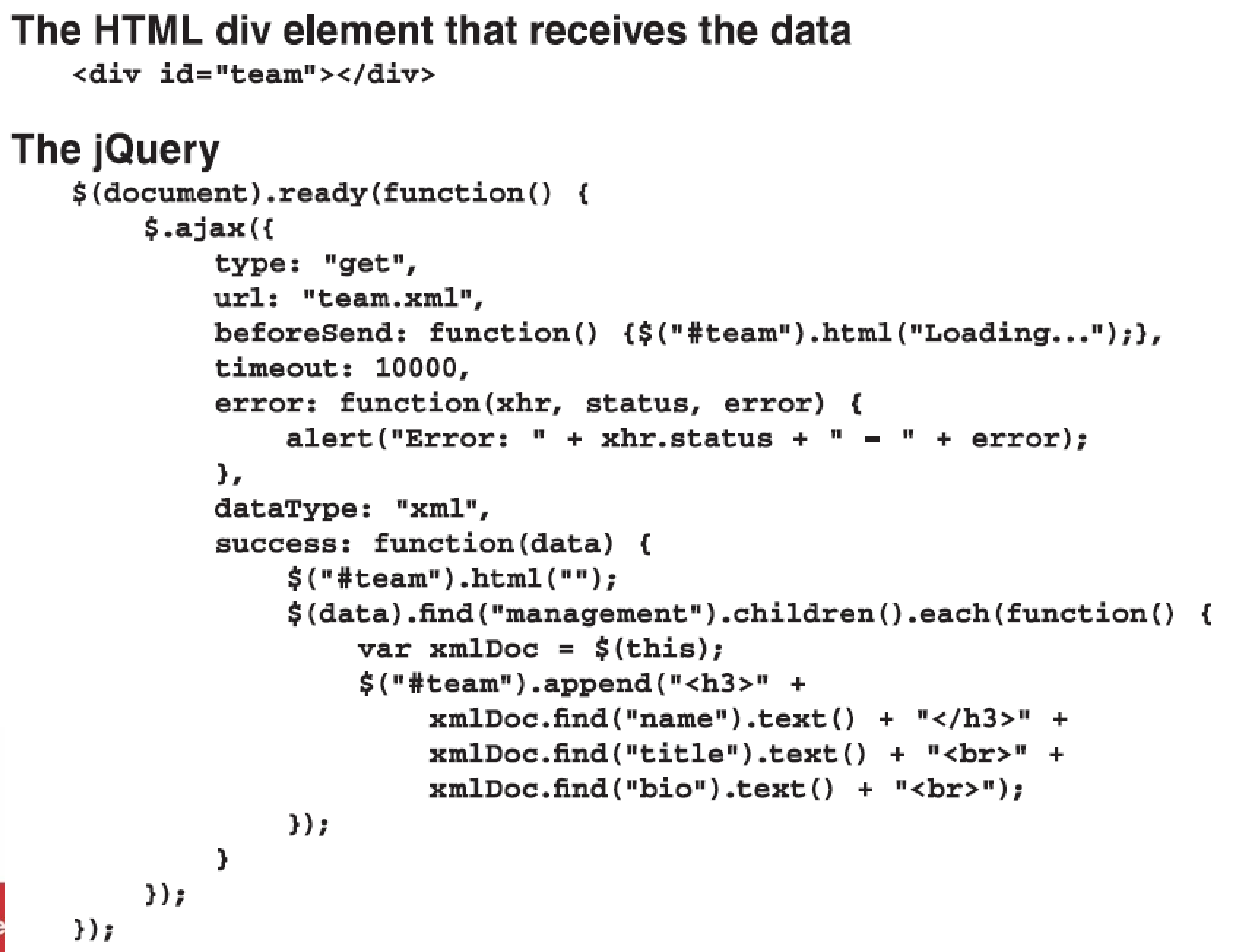
Loading...

Error: 404 - Not Found

The XML file

```
<management>
    <teammember>
        <name>Agnes</name>
        <title>Vice President of Accounting</title>
        <bio>With over 14 years of public accounting ... </bio>
    </teammember>
    ...
</management>
```

How to use the $.ajax() method to load data (cont.)



The HTML div element that receives the data

```
<div id="team"></div>
```

The jQuery

```
$(document).ready(function() {
    $.ajax({
        type: "get",
        url: "team.xml",
        beforeSend: function() {$("#team").html("Loading...");},
        timeout: 10000,
        error: function(xhr, status, error) {
            alert("Error: " + xhr.status + " - " + error);
        },
        dataType: "xml",
        success: function(data) {
            $("#team").html("");
            $(data).find("management").children().each(function() {
                var xmlDoc = $(this);
                $("#team").append("<h3>" +
                    xmlDoc.find("name").text() + "</h3>" +
                    xmlDoc.find("title").text() + "<br>" +
                    xmlDoc.find("bio").text() + "<br>");
            });
        }
    });
});
```

How to use Ajax with Flickr

How to use the feed API for Flickr

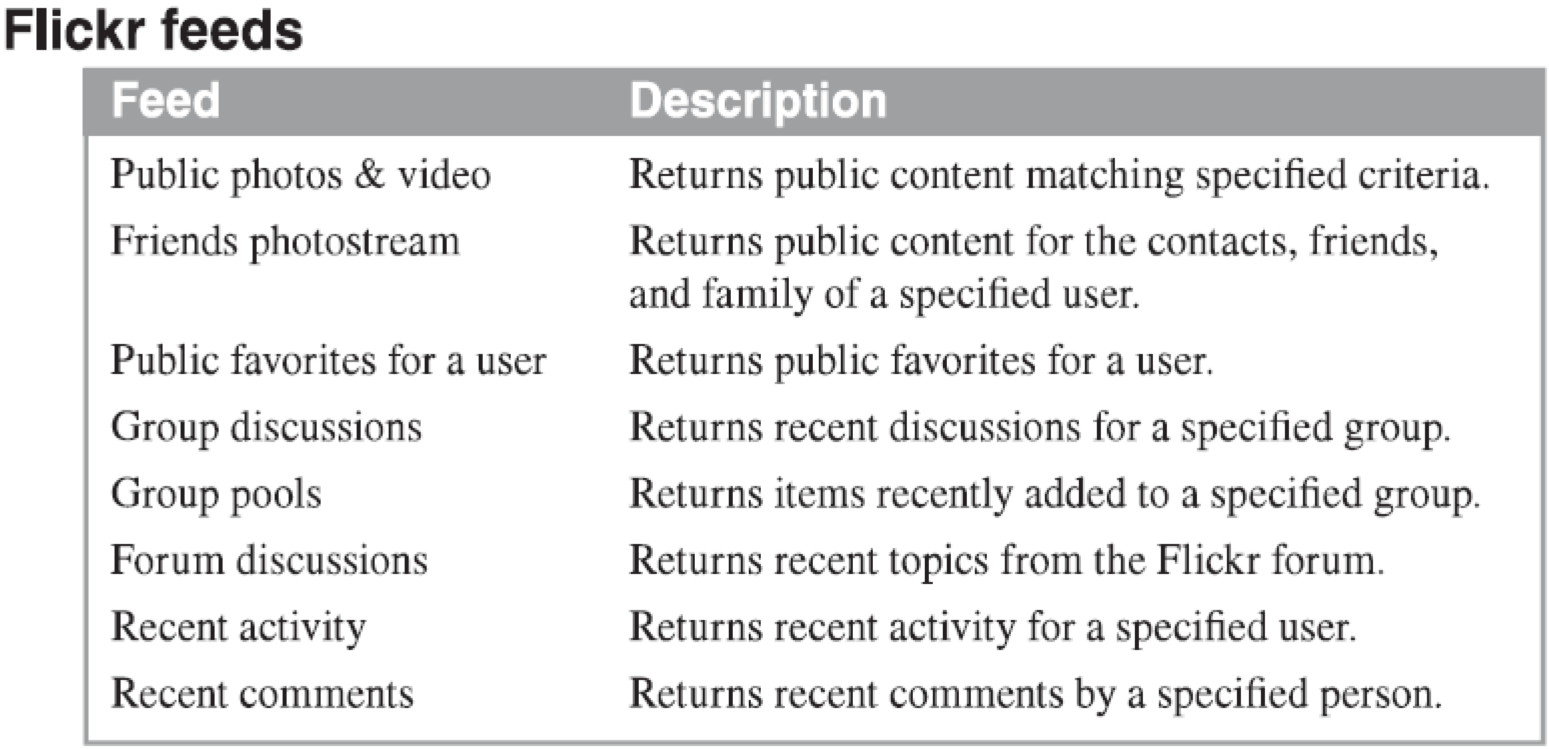
- Flickr is a website that lets you store your photos on it for free. That means you can access your photos whenever you are.
- Flickr provides a number of feeds that you can retrieve photos and related information from.
- The URL for the Flickr public feed document

www.flickr.com/services/feeds/docs/photo_public/

How to use the feed API for Flickr (cont.)

- The URL for the Flickr public feed document

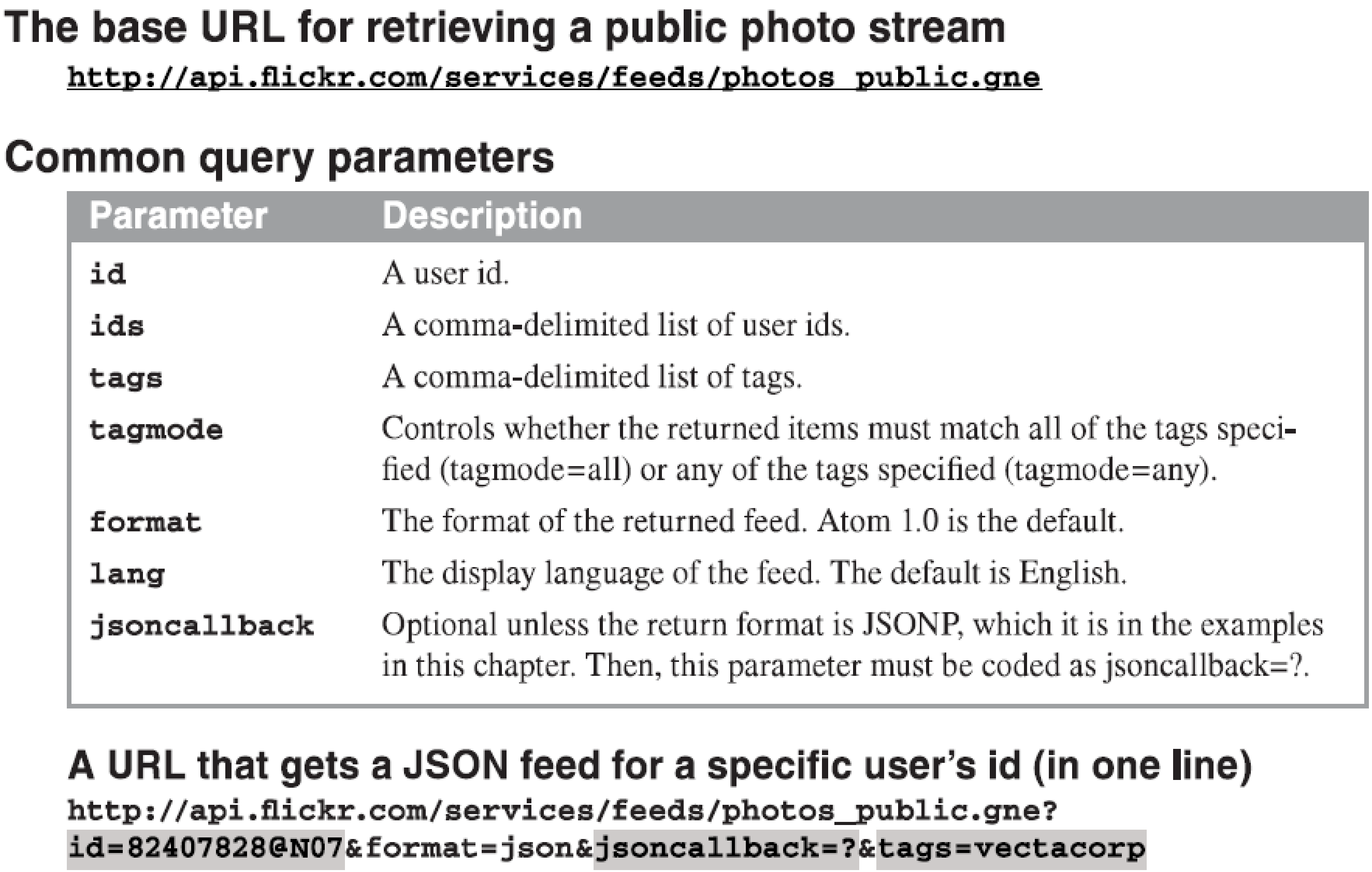
`www.flickr.com/services/feeds/docs/photo_public/`



Flickr feeds

Feed	Description
Public photos & video	Returns public content matching specified criteria.
Friends photostream	Returns public content for the contacts, friends, and family of a specified user.
Public favorites for a user	Returns public favorites for a user.
Group discussions	Returns recent discussions for a specified group.
Group pools	Returns items recently added to a specified group.
Forum discussions	Returns recent topics from the Flickr forum.
Recent activity	Returns recent activity for a specified user.
Recent comments	Returns recent comments by a specified person.

How to use the feed API for Flickr (cont.)



The base URL for retrieving a public photo stream

```
http://api.flickr.com/services/feeds/photos_public.gne
```

Common query parameters

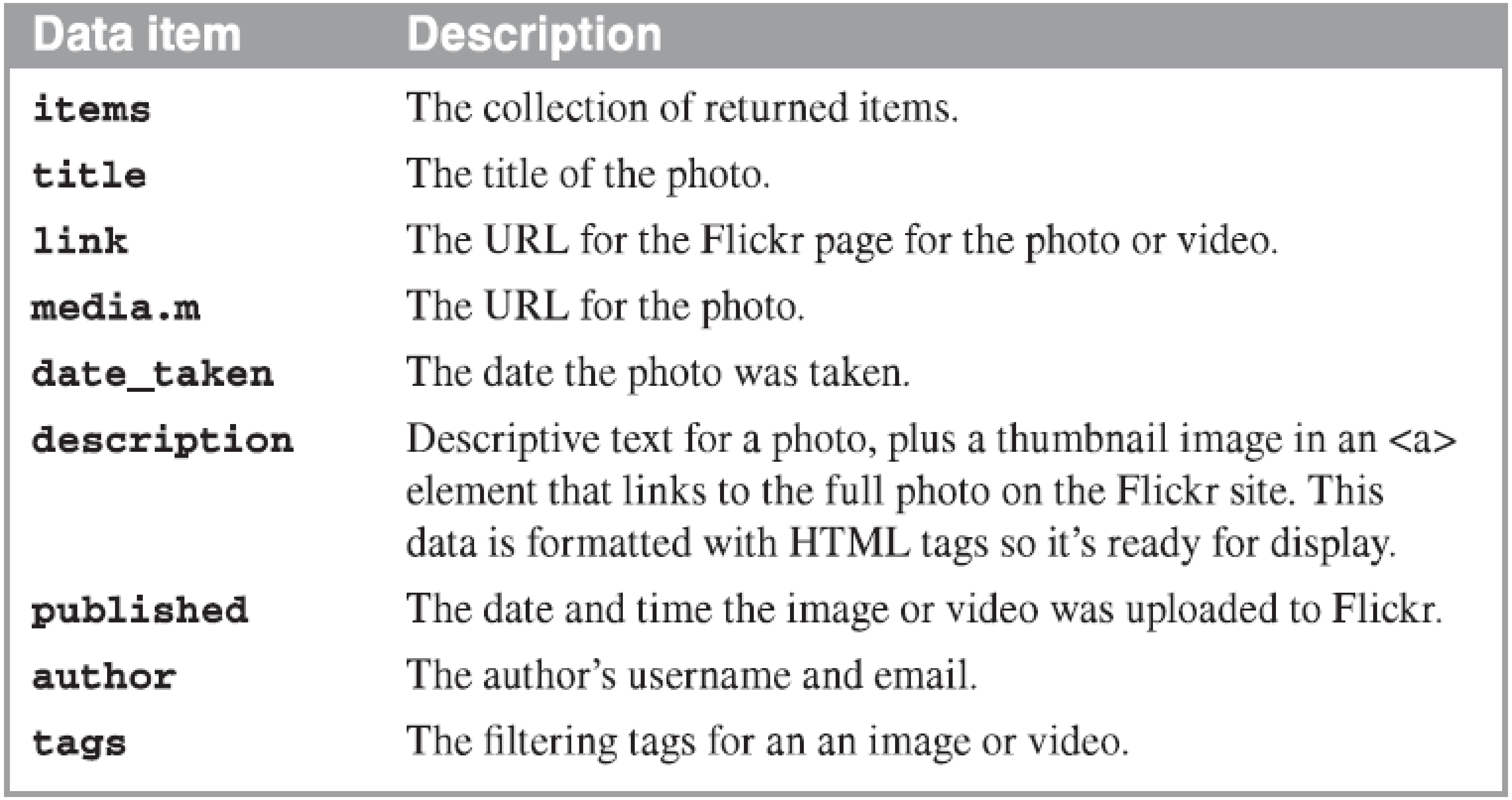
Parameter	Description
`id`	A user id.
`ids`	A comma-delimited list of user ids.
`tags`	A comma-delimited list of tags.
`tagmode`	Controls whether the returned items must match all of the tags specified (tagmode=all) or any of the tags specified (tagmode=any).
`format`	The format of the returned feed. Atom 1.0 is the default.
`lang`	The display language of the feed. The default is English.
`jsoncallback`	Optional unless the return format is JSONP, which it is in the examples in this chapter. Then, this parameter must be coded as jsoncallback=?.

A URL that gets a JSON feed for a specific user's id (in one line)

```
http://api.flickr.com/services/feeds/photos_public.gne?
id=82407828@N07&format=json&jsoncallback=?&tags=vectacorp
```

How to display Flickr data on a page

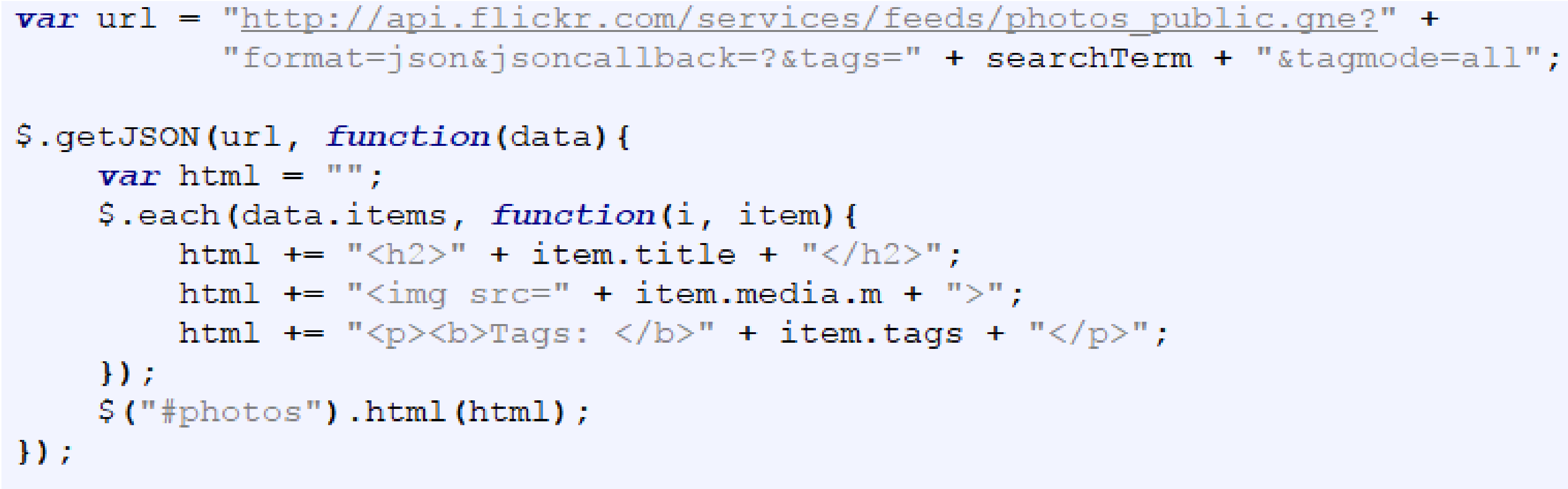
- Data items returned by a photo feed



Data item	Description
`items`	The collection of returned items.
`title`	The title of the photo.
`link`	The URL for the Flickr page for the photo or video.
`media.m`	The URL for the photo.
`date_taken`	The date the photo was taken.
`description`	Descriptive text for a photo, plus a thumbnail image in an <a> element that links to the full photo on the Flickr site. This data is formatted with HTML tags so it's ready for display.
`published`	The date and time the image or video was uploaded to Flickr.
`author`	The author's username and email.
`tags`	The filtering tags for an an image or video.

How to display Flickr data on a page (cont.)

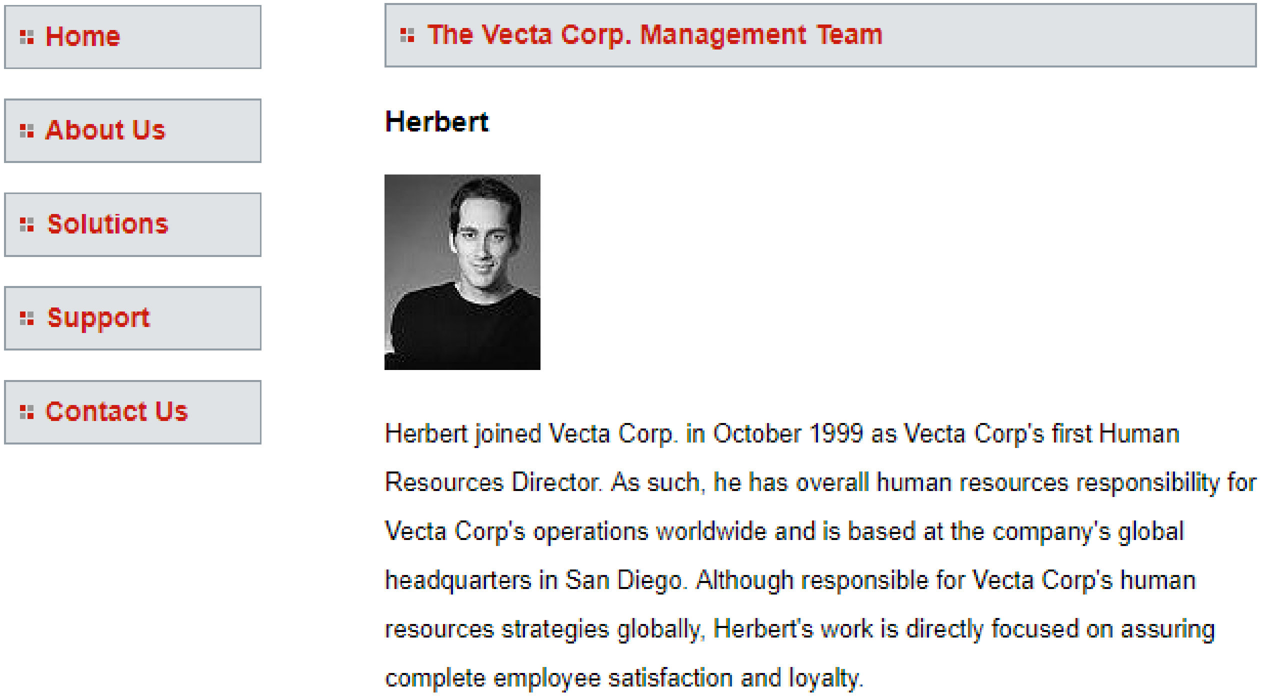
- jQuery code that gets the titles and photos from a photo feed



```
var url = "http://api.flickr.com/services/feeds/photos_public.gne?" +
          "format=json&jsoncallback=?&tags=" + searchTerm + "&tagmode=all";

$.getJSON(url, function(data){
    var html = "";
    $.each(data.items, function(i, item){
        html += "<h2>" + item.title + "</h2>";
        html += "<img src=" + item.media.m + ">";
        html += "<p><b>Tags: </b>" + item.tags + "</p>";
    });
    $("#photos").html(html);
});
```

How to display descriptions for a Flickr photo feed

- This application use the description data item to display the photo as well as the descriptive text.
- A web page that displays titles and descriptions for a photo feed:



The Vecta Corp. Management Team

Herbert

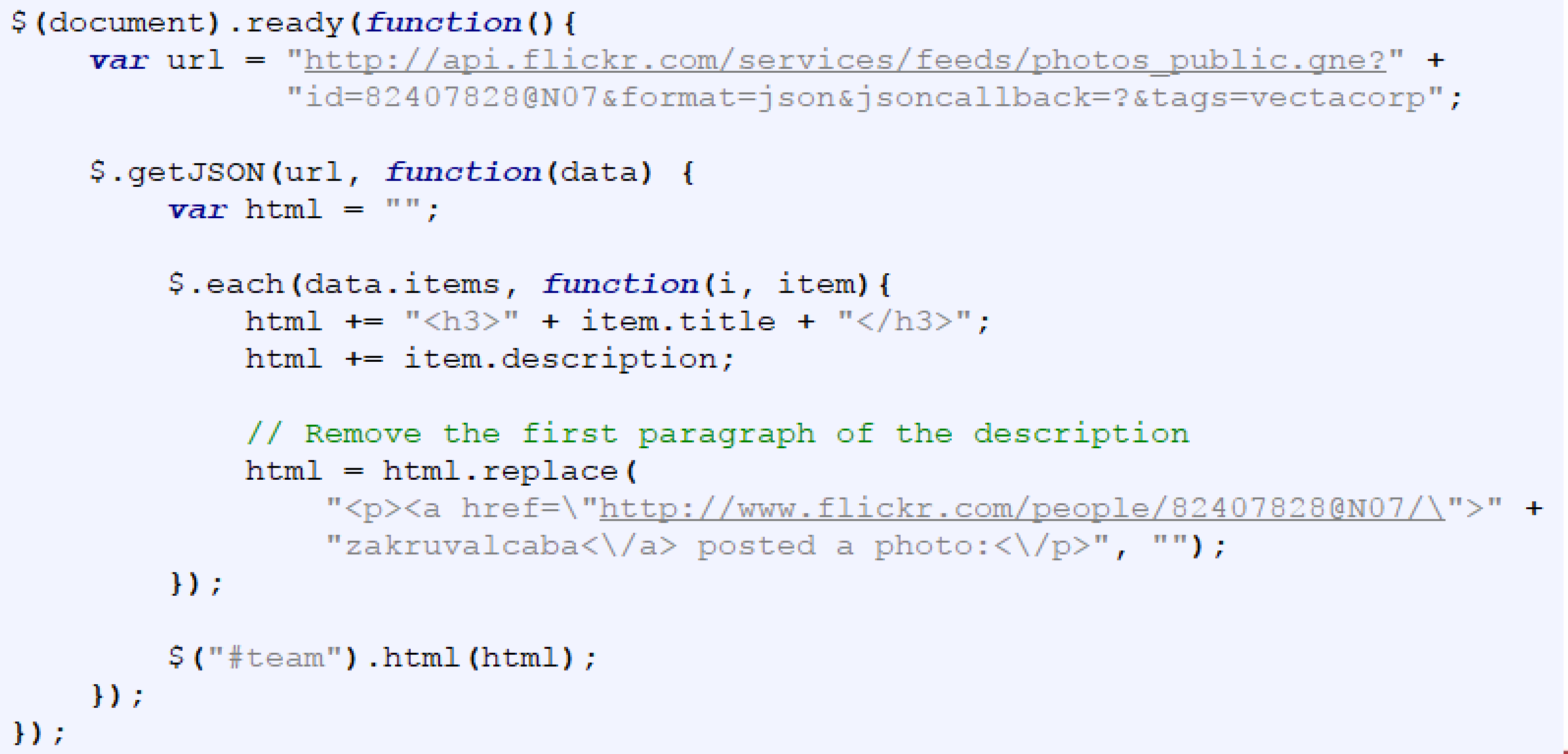
Herbert joined Vecta Corp. in October 1999 as Vecta Corp's first Human Resources Director. As such, he has overall human resources responsibility for Vecta Corp's operations worldwide and is based at the company's global headquarters in San Diego. Although responsible for Vecta Corp's human resources strategies globally, Herbert's work is directly focused on assuring complete employee satisfaction and loyalty.

How to display descriptions for a Flickr photo feed (cont.)

- The HTML element that will receive the data from the feed

  `<div id=“team”> </div>`

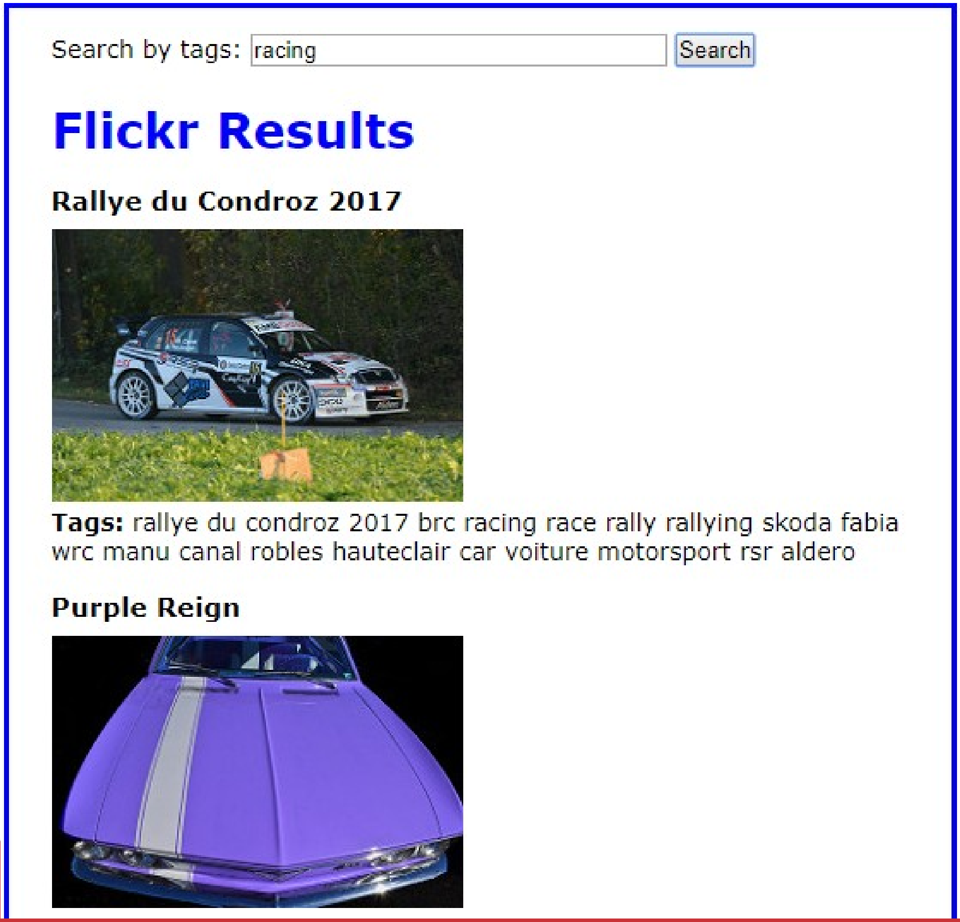
- The jQuery that retrieves and displays the data



```
$(document).ready(function(){
    var url = "http://api.flickr.com/services/feeds/photos_public.gne?" +
              "id=82407828@N07&format=json&jsoncallback=?&tags=vectacorp";

    $.getJSON(url, function(data) {
        var html = "";

        $.each(data.items, function(i, item){
            html += "<h3>" + item.title + "</h3>";
            html += item.description;

            // Remove the first paragraph of the description
            html = html.replace(
                "<p><a href=\"http://www.flickr.com/people/82407828@N07/\">" +
                "zakruvalcaba<\/a> posted a photo:<\/p>", "");
        });

        $("#team").html(html);
    });
});
```

How to search for photos by tags

- This application permit you search photos from Flickr by tags like “racing”, “birthday”..



Search by tags: racing

Flickr Results

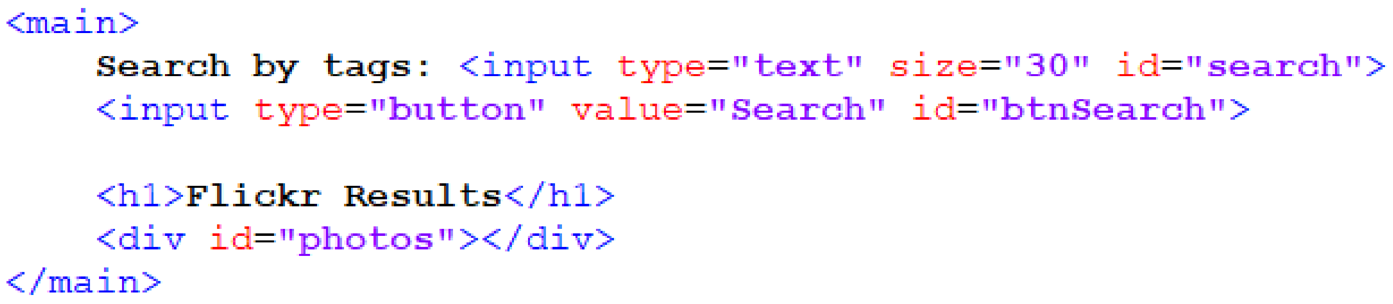
Rallye du Condroz 2017

Tags: rallye du condroz 2017 brc racing race rally rallying skoda fabia wrc manu canal robles hauteclair car voiture motorsport rsr aldero

Purple Reign

How to search for photos by tags (cont.)

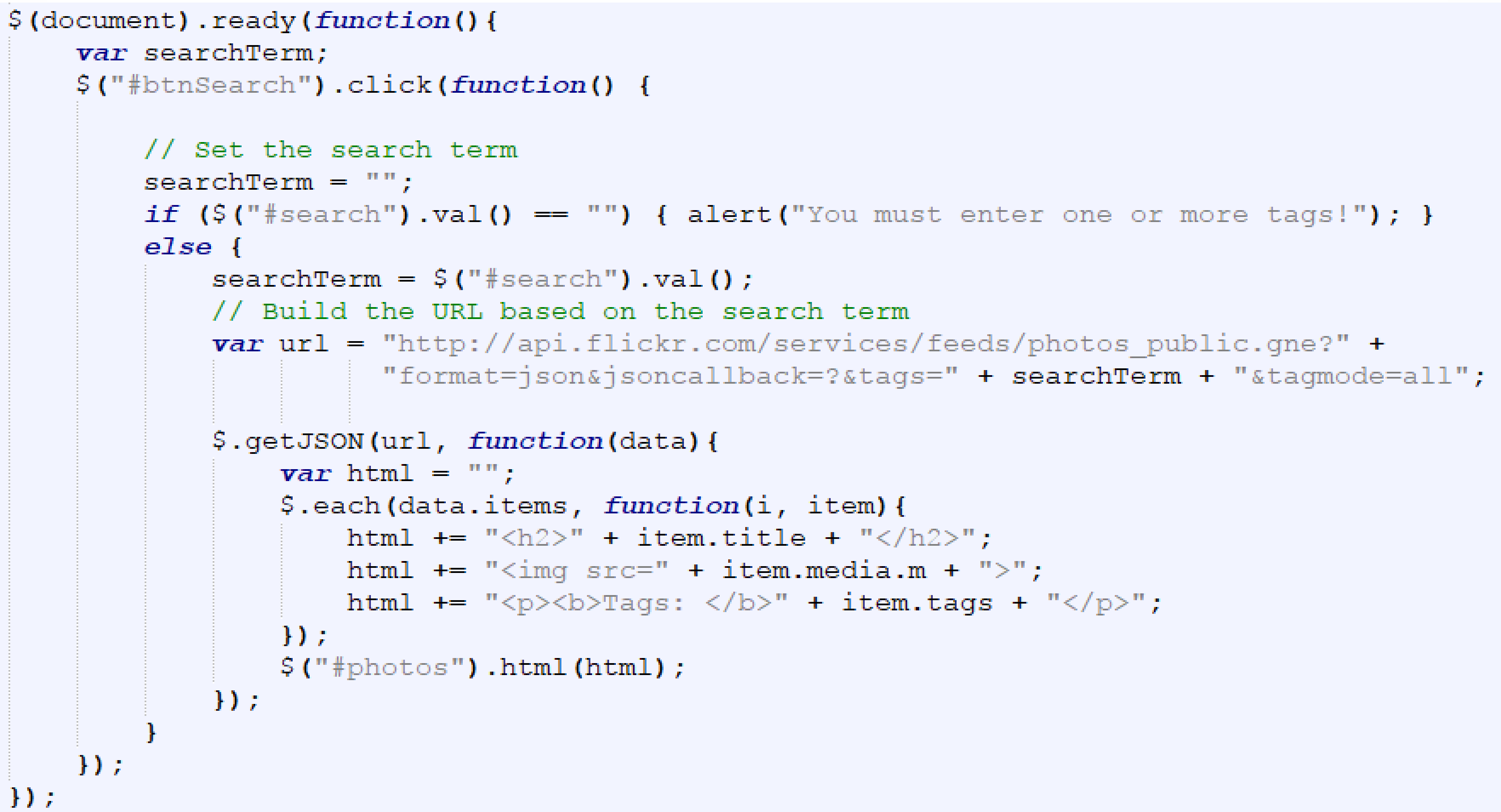
- HTML code:



```
<main>
    Search by tags: <input type="text" size="30" id="search">
    <input type="button" value="Search" id="btnSearch">

    <h1>Flickr Results</h1>
    <div id="photos"></div>
</main>
```

How to search for photos by tags (cont.)

- JavaScript code:



```
$(document).ready(function(){
    var searchTerm;
    $("#btnSearch").click(function() {

        // Set the search term
        searchTerm = "";
        if ($("#search").val() == "") { alert("You must enter one or more tags!"); }
        else {
            searchTerm = $("#search").val();
            // Build the URL based on the search term
            var url = "http://api.flickr.com/services/feeds/photos_public.gne?" +
                      "format=json&jsoncallback=?&tags=" + searchTerm + "&tagmode=all";

            $.getJSON(url, function(data){
                var html = "";
                $.each(data.items, function(i, item){
                    html += "<h2>" + item.title + "</h2>";
                    html += "<img src=" + item.media.m + ">";
                    html += "<p><b>Tags: </b>" + item.tags + "</p>";
                });
                $("#photos").html(html);
            });
        }
    });
});
```

Summary

- ***Ajax*** stands for *Asynchronous JavaScript and XML*.
- Unlike normal HTTP request, Ajax let you receive data from web server without reloading the page.
- Ajax need to use JavaScript to send the request, process the responses and updates the DOM with the new data.
- To send an Ajax request, JavaScript uses a browser object known as XMLHttpRequest (XHR) object.
- jQuery includes several shorthand methods that let you request and receive HTML, XML or JSON data.
- When you send data with an Ajax request, the URL is for a server-side script such as a PHP file. Then, the script is responsible for returning the data in XML or JSON format.

Summary (cont.)

- The **$.ajax()** method provides options that give you more control over the way the Ajax request works, such as providing a function for handling errors.
- The **jqXHR** object is jQuery’s superset of the standard XMLHttpRequest object that provide properties of that project.
- Flickr is a website that lets you store your photos on it for free. That means you can access your photos whenever you are.
- Flickr provides a number of feeds that you can retrieve photos and related information from.

The End.