

Any good AJAX framework for Google App Engine apps? [closed]

Asked 16 years, 3 months ago Modified 13 years, 5 months ago

Viewed 12k times  Part of [Google Cloud](#) Collective



14



Closed. This question is seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. It does not meet [Stack Overflow guidelines](#). It is not currently accepting answers.



We don't allow questions seeking recommendations for software libraries, tutorials, tools, books, or other off-site resources. You can edit the question so it can be answered with facts and citations.

Closed 5 years ago.

[Improve this question](#)

I am trying to implement AJAX in my Google App Engine application, and so I am looking for a good AJAX framework that will help me. Anyone has any idea?

I am thinking about Google Web Toolkit, how good it is in terms of creating AJAX for Google App Engine?



python

ajax

google-app-engine

Share

Improve this question

Follow

asked Sep 10, 2008 at 13:12



Graviton

83.2k ● 147 ● 439 ● 611

11 Answers

Sorted by:

Highest score (default)



12



As Google Web Toolkit is a [subset of Java](#) it works best when you Java at the backend too. Since Google App Engine is currently [Python only](#) I think you'd have to do a lot of messing about to get your server and client to talk nicely to each other.



jQuery seems to be the most popular JavaScript library option in the [AJAX Tag at DjangoSnippets.com](#).



Edit: The above is only true of Google App Engine applications written in Python. As Google App Engine now supports Java, GWT could now be a good choice for writing an AJAX front end. [Google even have a tutorial showing you how to do it.](#)

Share Improve this answer

edited Jan 5, 2010 at 5:02

Follow

answered Sep 10, 2008 at 13:14



David Webb

193k ● 57 ● 316 ● 302

Thanks Dave. Ah, I suppose that this rules out the potential of using Google Web Toolkit as the AJAX framework for GAE. Too bad! Is there any other framework? – [Graviton](#) Sep 10, 2008 at 13:19



7



A nice way is to use an AJAX library is to take advantage of [Google's AJAX Libraries API service](#). This is a bit faster and cleaner than downloading the JS and putting it in your `/static/` folder and doesn't eat into your disk quota.

In your javascript you would just put, for example:

```
google.load("jquery", "1.3.2");
```

and/or

```
google.load(google.load("dojo", "1.3.0");
```

Somewhere in your header you would put something like:

```
<script src="http://www.google.com/jsapi?key=your-key-
```

And that's all you need to use Google's API libraries.

Share Improve this answer

answered Apr 28, 2009 at 22:08

Follow



Scott Kirkwood

1,135 ● 12 ● 15



4

Here is how we've implemented Ajax on the Google App Engine, but the idea can be generalized to other platforms.



We have a handler script for Ajax requests that responds -mostly- with JSON responses. The structure looks something like this (this is an excerpt from a standard GAE handler script):

```
def Get(self, user):
    self.handleRequest()

def Post(self, user):
    self.handleRequest()

def handleRequest(self):
    '''
    A dictionary that maps an operation name to a command
    aka: a dispatcher map.
    '''
    operationMap = {'getfriends': [GetFr
                        'requestfriend': [Reque
[self.request.get('id')]],
                        'confirmfriend': [Confi
[self.request.get('id')]],
                        'ignorefriendrequest': [Ignor
[self.request.get('id')]],
                        'deletefriend': [Delet
[self.request.get('id')]]]

    # Delegate the request to the matching command cla
```

The commands are a simple implementation of the command pattern:

```
class Command():
    """ A simple command pattern.
    """
    _valid = False
    def validate(self):
        """ Validates input. Sanitize user input here.
        """
```

```

        self._valid = True

    def _do_execute(self):
        """ Executes the command.
            Override this in subclasses.
        """
        pass

    @property
    def valid(self):
        return self._valid

    def execute(self):
        """ Override _do_execute rather than this.
        """
        try:
            self.validate()
        except:
            raise
        return self._do_execute()

# Make it easy to invoke commands:
# So command() is equivalent to command.execute()
__call__ = execute

```

On the client side, we create an Ajax delegate. Prototype.js makes this easy to write and understand. Here is an excerpt:

```

/**
 * Ajax API
 *
 * You should create a new instance for every call.
 */
var AjaxAPI = Class.create({
    /* Service URL */
    url: HOME_PATH+"ajax/",

    /* Function to call on results */
    resultCallback: null,

```

```

/* Function to call on faults. Implementation not
faultCallback: null,

/* Constructor/Initializer */
initialize: function(resultCallback, faultCallback)
    this.resultCallback = resultCallback;
    this.faultCallback = faultCallback;
},

requestFriend: function(friendId){
    return new Ajax.Request(this.url + '?op=requestFriend',
    {method: 'post',
    parameters: {'id': friendId},
    onComplete: this.resultCallback
    });
},

getFriends: function(){
    return new Ajax.Request(this.url + '?op=getFriends',
    {method: 'get',
    onComplete: this.resultCallback
    });
}

});

```

to call the delegate, you do something like:

```

new AjaxApi(resultHandlerFunction, faultHandlerFunction)

```

I hope this helps!

Share Improve this answer

Follow

answered Mar 3, 2009 at 4:56



mahmoud

595 ● 2 ● 6



4



There is no reason why you shouldn't use GAE and Google Web Toolkit (GWT) together. You write your backend code in Python and the frontend code in Java (and possibly some JavaScript), which is then compiled to JavaScript. When using another AJAX framework you will also have this difference between server and client side language.

GWT has features that make remote invocation of java code on the server easier, but these are entirely optional. You can just use JSON or XML interfaces, just like with other AJAX frameworks.

GWT 1.5 also comes with JavaScript Overlay Types, that basically allow you to treat a piece of JSON data like a Java object when developing the client side code. You can read more about this [here](#).

Update:

Now that Google has added Java support for Google App Engine, you can develop both backend and frontend code in Java on a full Google stack - if you like. There is a nice [Eclipse plugin](#) from Google that makes it very easy to develop and deploy applications that use GAE, GWT or both.

Share Improve this answer

edited Jun 25, 2009 at 18:57

Follow

answered Sep 10, 2008 at 19:00



Christian Berg

14.5k ● 9 ● 41 ● 44



3



I'd recommend looking into a pure javascript framework (probably JQuery) for your client-side code, and write JSON services in python- that seems to be the easiest / bestest way to go.

Google Web Toolkit lets you write the UI in Java and compile it to javascript. As Dave says, it may be a better choice where the backend is in Java, as it has nice RPC hooks for that case.

Share Improve this answer

Follow

answered Sep 10, 2008 at 13:19



Tim Howland

7,970 ● 4 ● 29 ● 46



2



You may want to have a look at Pyjamas (<http://pyjs.org/>), which is "GWT for Python".

Share Improve this answer

Follow

answered Dec 9, 2008 at 8:33



fabs



2



try also [GQuery for GWT](#). This is Java code:

```
public void onModuleLoad() {
    $("div").css("color", "red").click(new Function()
        public void f(Element e) {
            Window.alert("Hello");
            $(e).as(Effects).fadeOut();
        }
    });
}
```

Being Java code resulting in somewhat expensive compile-time (Java->JavaScript) optimizations and easier refactoring.

Nice, it isn't?

Share Improve this answer

answered Jun 15, 2009 at 8:12

Follow



dfa

116k ● 32 ● 192 ● 229



1



jQuery is a fine library, but also check out the [Prototype JavaScript framework](#). It really turns JavaScript from being an occasionally awkward language into a beautiful and elegant language.

Share Improve this answer

answered Sep 10, 2008 at 13:29

Follow



Christian Oudard

49.8k ● 25 ● 70 ● 70



0



If you want to be able to invoke method calls from JavaScript to Python, [JSON-RPC](#) works well with Google App Engine. See Google's article, "[Using AJAX to Enable Client RPC Requests](#)", for details.

Share Improve this answer

edited Sep 15, 2008 at 13:49



Follow



answered Sep 15, 2008 at 13:43



Steve

684 ● 7 ● 8



0



I'm currently using JQuery for my GAE app and it works beautifully for me. I have a chart (google charts) that is dynamic and uses an Ajax call to grab a JSON string. It really seems to work fine for me.

Share Improve this answer

answered Oct 23, 2008 at 16:41



Follow



o2bjang

35 ● 5



0



Google has recently announced the Java version of Google App Engine. This release also provides an Eclipse plugin that makes developing GAE applications with GWT easier.

See details here:

<http://code.google.com/appengine/docs/java/overview.html>



Of course, it would require you to rewrite your application in Java instead of python, but as someone who's worked with GWT, let me tell you, the advantages of using a modern IDE on your AJAX codebase are totally worth it.

Share Improve this answer

answered Apr 28, 2009 at 22:17

Follow



levik

117k ● 28 ● 75 ● 92
