Quanah Documentation

Release 0.3.0

Sean Wilkinson

Contents

1 Overview					
	1.1 How it works				
	1.2 How to use it				
	1.3 How to contribute				
	How to use it 2.1 Loading the module				
3	Source code				

Overview

Quanah is a model for workflows, and it is so closely related to its original implementation in JavaScript as to be synonymous. Quanah is the core model that powers QMachine (QM), a web service for distributed computing. QM was detailed in a recent paper, QMachine: commodity supercomputing in web browsers, but Quanah was only briefly mentioned. This manual is the best reference for Quanah that is currently available. The main focuses here will be to describe how it works, how to use it, and how to contribute to the open-source project.

1.1 How it works

(coming soon)

1.2 How to use it

See How to use it (page 3).

1.3 How to contribute

See Source code (page 5).

How to use it

2.1 Loading the module

Quanah is versatile enough to run in any JavaScript environment, but it can be tricky to figure out how to do it in a given environment. While by no means exhaustive, the following directions cover most of the common environments. The main idea is to demonstrate how to load Quanah and store a reference to it as quanah.

2.1.1 ArangoDB shell

```
var quanah = require("quanah");
```

2.1.2 Google D8 / V8 shells

```
load("quanah.js");
var quanah = QUANAH;
```

2.1.3 JavaScriptCore shell

```
load("quanah.js");
var quanah = QUANAH;
```

2.1.4 jrunscript

```
load("quanah.js");
var quanah = QUANAH;
```

2.1.5 MongoDB shell

```
load("quanah.js");
var quanah = QUANAH;
```

2.1.6 Mozilla Rhino

```
load("quanah.js");
var quanah = QUANAH;
```

2.1.7 Mozilla Spidermonkey shell

```
load("quanah.js");
var quanah = QUANAH;
```

2.1.8 Node.js

Node.js is a server-side JavaScript environment for writing high-performance networking applications, and it is arguably the greatest recent source for innovation in asynchronous programming outside of the web browser.

```
var quanah = require("./quanah");
```

2.1.9 PhantomJS

```
var quanah = require("./quanah");
```

2.1.10 RingoJS

```
var quanah = require("./quanah");
```

2.1.11 Web browsers

Browsers can be the easiest or hardest to load external code with, depending on the strategy that you choose. The easiest route is to add one line to an HTML page:

```
<script src="quanah.js"></script>
```

Then, the programs that use Quanah should be loaded in the same way, by lines that occur later in the page. To reference Quanah, those programs only need to contain the following:

```
var quanah = QUANAH;
```

To load Quanah dynamically is more complicated, and it demonstrates nicely why Quanah itself is so useful:

```
var script = document.createElement("script");
script.onload = function () {
    var quanah = QUANAH;
    // (your code goes here)
};
script.src = "quanah.js";
document.body.appendChild(script);
```

The problem with the second strategy is that it requires some extra knowledge about asynchronous programming, which is precisely what Quanah was originally designed to simplify.

СН	۸	D٦	ΓE	R	3
υп	н	Г		п	J

Source code

Quanah's source code is available at https://github.com/qmachine/quanah.