ajax event system

As you may have seen with the ajax div template and ajax a template, the framework and Dojo provide a nice way to subscribe and notify of topics from within the browser. A benifit of using Dojo as the basis of many of these components is being able to loosely couple UI components. There are two attributes of importance: `listenTopics` and `notifyTopics`.

- If a component has a `notifyTopics` attribute, then after the processing has been completed a message with be published to the topic names supplied as a value (comma delimited).
- If a component has a `listenTopics` attribute, then when a message is published to the topic names supplied as a value (comma delimited), the component will perform custom tag-specific logic (i.e a DIV tag will re-fresh its content).

As well as this, you can publish and subscribe to topic names with javascript code. To publish to the `topic_name` topic:

```
dojo.event.topic.publish("topic_name", "content");
```

The topic_name attribute is required, the content attribute is not and most elements are triggered without having this attribute. See the ajax div template for an example of this type of interaction.

To subscribe to the *topic_name* topic:

```
function doSomethingWithEvent(data) {
...
}

dojo.event.topic.getTopic("topic_name").subscribe(null, "doSomethingWithEvent");
```

The `subscribe` method takes 2 parameters, the first is the JavaScript object variable (or null if the function is not from an object) and the second is the name of the function to call when an event is recieved on the topic.