**Tracker Ajax Services**

Some significant developments have taken place since tiki 6 to make development with trackers much easier. tiki-ajax_services.php has been designed to create, update, and delete trackers, tracker fields, and tracker items. This is a primer on the available services and how they can be used.

A jQuery plugin has been written for tiki 10 (trunk at time of writing) that is accessible in /lib/jquery_tiki/tiki-trackers.js and can be added to a php page after including tiki-setup.php like this:

```php
TikiLib::lib('header')->add_jsfile('lib/jquery_tiki/tiki-trackers.js');
```

This above file sets many of the standards for how we can use tiki-ajaxservices.php, however it can be called directly, and is in many cases.

**Methods**

Methods found in tiki-trackers.js that enhance the ability to use trackers as an api for building applications:

**Method Options**

The method options used in the below functions are very similar, this is the description of them so it isn't repeated below:

- `trackerId`: int, default = 0, optional if trackerName is set
- `trackerName`: string, default = '', optional if trackerId is null or 0
- `itemId`: int, if set to 0, default = 0, inputs return blank or blanks with defaults
- `byName`: bool, default = false, if set to true field keys become field names and can use either trackerId or trackerName
- `defaults`: object, default = {}, if set key = field key, value = default value for a new item/itemId (0)

**Method**: `.tracker_get_item_inputs(options, callBackFunction);`

Description: Gets the tracker value in their field inputs in an array, each value being html that should be wrapped in another html element (ie td, div, span) where the key of the object is the fieldId or fieldName (depending if the byName option is set to true). The input names are set to what is expected from

Options:

- `trackerId`
- trackerName
- itemId
- byName
- defaults

Example 1:

```javascript
var inputs = {};

$.tracker_get_item_inputs({
    trackerId: 1,
    defaults: {
        '1': 'Foo', // 1 is the id of a field
        '2': 'Foo 2' // 2 is the id of a field
    },
    function(o) {
        inputs = o;
    }
});
```

Example 2:

```javascript
var inputs = {};

$.tracker_get_item_inputs({
    trackerName: 'My Tracker',
    byName: true,
    defaults: {
        'Field 1': 'Foo', // Field 1 is the name of a field
    }
});
```
'Field 2': 'Foo 2' //2 is the name of a field

}, function(o) {

  inputs = o;

});

Example 3:

var inputs = {};

$.tracker_get_item_inputs({

  trackerName: 'My Tracker',

  byName: true,

  itemId: 100 //if item id is set, then the inputs are populated with the values from that tracker item's field's values

}, function(o) {

  inputs = o;

});

Method: $.fn.tracker_get_inputs_from_form();

Description: Gets the fields from the form they have been added to.

Example:

var inputs = $('form').tracker_get_inputs_from_form();

Method: $.fn.tracker_insert_item(options, callBackFunction);

Description: Inserts a new item into a tracker from a form that has inputs appended from $.tracker_get_item_inputs and afterward performs a callback.

Options:
• trackerId
• trackerName
• byName
• fields

Example:

```javascript
$('form').tracker_insert_item({
  trackerId: 5,
  fields: $('form').tracker_get_inputs_from_form()
}, function(trackerInfo) {
  document.location = document.location + '';
});
```

**Method:** $.fn.tracker_remove_item(options, callBackFunction);

**Description:** removes/deletes an item from a tracker

**Options:**

• trackerId
• trackerName
• itemId

**Method:** $.fn.tracker_update_item(options, callBackFunction);

**Description:** Updates a tracker item's fields

**Options:**

• trackerId
• trackerName
- itemId
- byName
- fields

Example:

```javascript
$('form').tracker_insert_item({
  trackerName: 'My Tracker',
  byName: true,
  itemId: 100,
  fields: $('form').tracker_get_inputs_from_form()
}, function(trackerInfo) {
  document.location = document.location + "";
});
```