# Ideas for budget-rest

* Offline use of application / localstorage and sync data with websocket
* Security
* Budget per user

## PushState for real restfull url’s

<https://github.com/tbranyen/backbone-boilerplate/blob/master/app/main.js>

Backbone.history.start({ pushState: true, root: app.root });

// All navigation that is relative should be passed through the navigate

// method, to be processed by the router. If the link has a `data-bypass`

// attribute, bypass the delegation completely.

$(document).on("click", "a[href]:not([data-bypass])", function(evt) {

// Get the absolute anchor href.

var href = { prop: $(this).prop("href"), attr: $(this).attr("href") };

// Get the absolute root.

var root = location.protocol + "//" + location.host + app.root;

// Ensure the root is part of the anchor href, meaning it's relative.

if (href.prop.slice(0, root.length) === root) {

// Stop the default event to ensure the link will not cause a page

// refresh.

evt.preventDefault();

// `Backbone.history.navigate` is sufficient for all Routers and will

// trigger the correct events. The Router's internal `navigate` method

// calls this anyways. The fragment is sliced from the root.

Backbone.history.navigate(href.attr, true);

}

});

Your server must then be able to serve those pages!!

**pushState** support exists on a purely opt-in basis in Backbone. Older browsers that don't support pushState will continue to use hash-based URL fragments, and if a hash URL is visited by a pushState-capable browser, it will be transparently upgraded to the true URL. Note that using real URLs requires your web server to be able to correctly render those pages, so back-end changes are required as well. For example, if you have a route of /documents/100, your web server must be able to serve that page, if the browser visits that URL directly. For full search-engine crawlability, it's best to have the server generate the complete HTML for the page ... but if it's a web application, just rendering the same content you would have for the root URL, and filling in the rest with Backbone Views and JavaScript works fine.