

## Redmine - Feature #31864

### Issue/TimeEntry API does not report root\_id

2019-08-07 17:14 - Adrian Fey

<b>Status:</b> New	<b>Start date:</b>
<b>Priority:</b> Normal	<b>Due date:</b>
<b>Assignee:</b>	<b>% Done:</b> 0%
<b>Category:</b> REST API	<b>Estimated time:</b> 0.00 hour
<b>Target version:</b>	
<b>Resolution:</b>	
<b>Description</b>	
<p>I would expect, that the Issues API exposes all fields, which I can find in the corresponding tables when accessing the database directly.</p> <p>For my current project I am missing the root_id field, which I require to build a report which indents child issues.</p> <p>Use case: I want to make a report about all Tickets, that have been worked on during a time period. Currently my approach would be:</p> <ol style="list-style-type: none"><li>1. Get all TimeEntrys with Spent_on in my timeframe</li><li>2. Get all Issues for those TimeEntrys</li><li>3. Truncate all Issues where issue_id &lt;&gt; root_id (can't be done currently, because information is not provided via API)</li><li>4. For all the root issues then again grab all children and build a tree</li></ol> <p>This by far isn't ideal request count wise, but everything else would require to have the root_issue_id information also on TimeEntrys.</p> <p>If TimeEntrys would also provide root_id of the issue, then I could simply:</p> <ol style="list-style-type: none"><li>1. Get all TimeEntrys with Spent_on between my timeframe</li><li>2. Concat all root_ids to a comma separated string and retrieve all root issues in one request</li><li>3. recursively grab the children, if existing, and build the tree</li></ol> <p>An even better approach might be to extend the Api for Issues to Include TimeEntrys and vice versa. That way one would be very flexible in making reports, which depend on time_spent during periods.</p>	