

在Obsidian中做时间记录复盘

#writing

大家好我是xy，今天给大家分享一个把Toggl导入Obsidian进行复盘的工作流。虽然社区插件里有Toggl插件，但是因为不支持手机端，无法融入我平时的工作流，所以就自己利用Toggl的官方API写了一个脚本。

功能概述：

用到的工具：

- Toggl - 时间记录
- Obsidian插件
 - Templater (>=1.14.0) - 获取Toggl记录并记入每日笔记
 - Tracker - 时间使用可视化

使用效果

每日笔记中可以通过Templater插入最近24小时的时间使用情况。会有每个大类所花的总时间以及每个小类所花的时间。具体如下：

🕒 时间记录 23:22 >

休息:: 8.3
学习:: 0.7
日常:: 1.3
运动:: 0.3
工作:: 7.5
睡觉:: 7.8

【休息】 --标签-描述: 7hr24; 刷刷刷-油管: 55min. Total: 8hr19.

【学习】 --书法-练字: 41min.

【日常】 --吃-午饭: 1hr; 做饭-做午饭: 18min. Total: 1hr18.

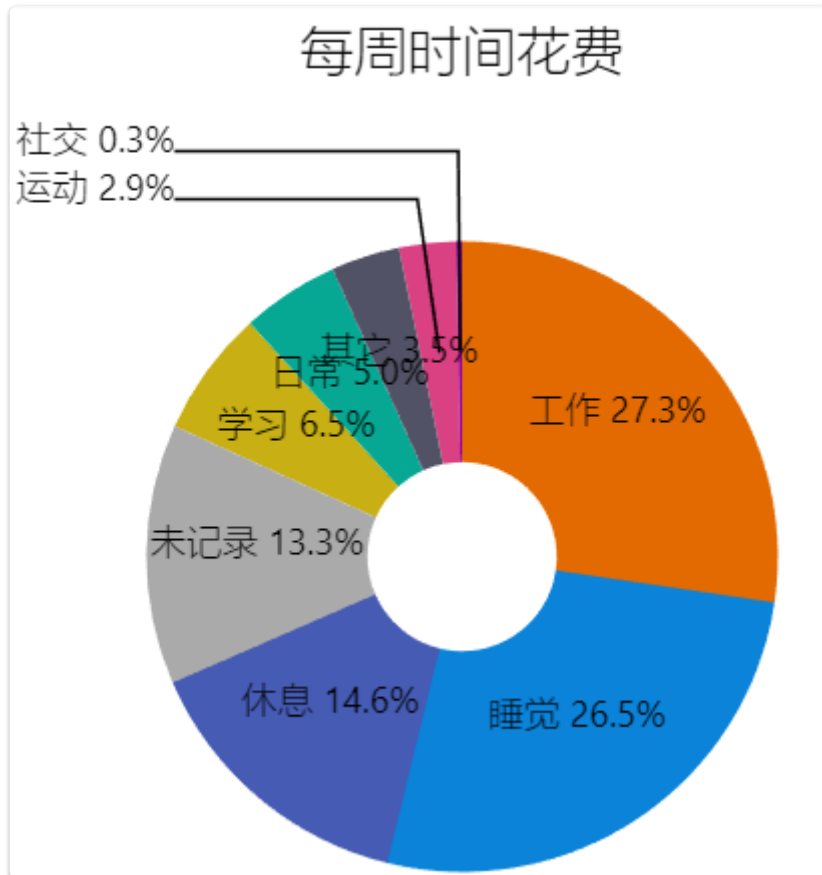
【运动】 --20min.

【工作】 --7hr31.

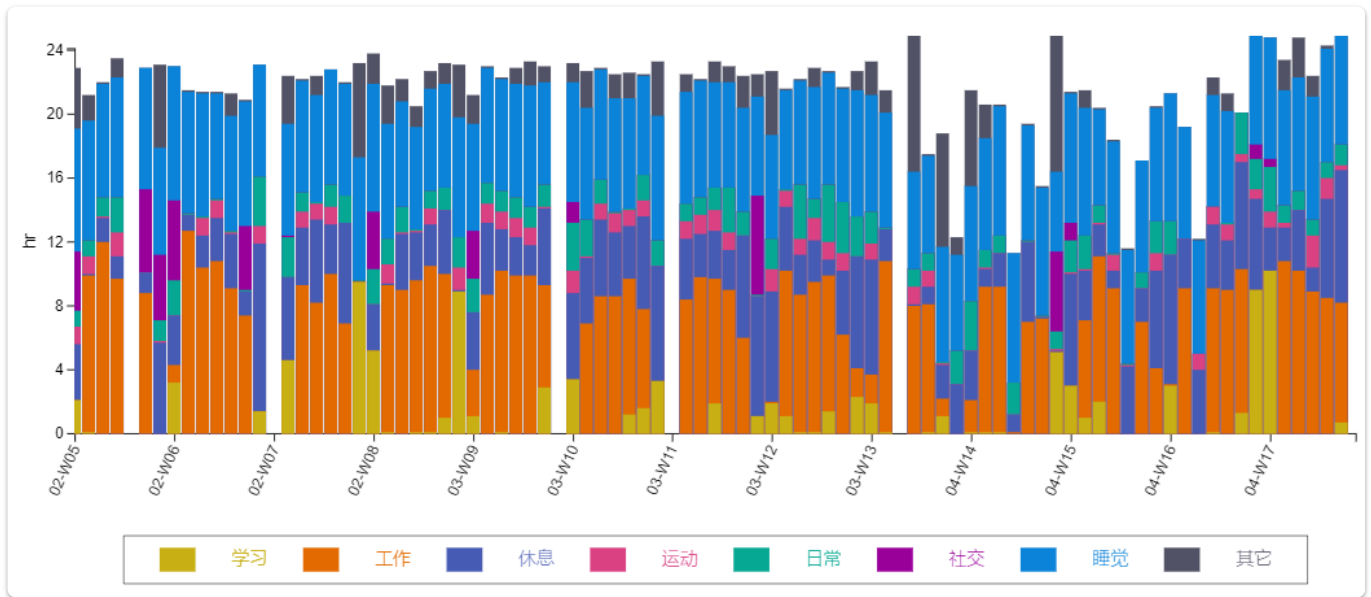
【睡觉】 --7hr50.

因为在时间记录信息里有每个项目的花费时间，所以可以用来生成各式图表。

比如**每周时间花费饼图**



季度时间花费分布图

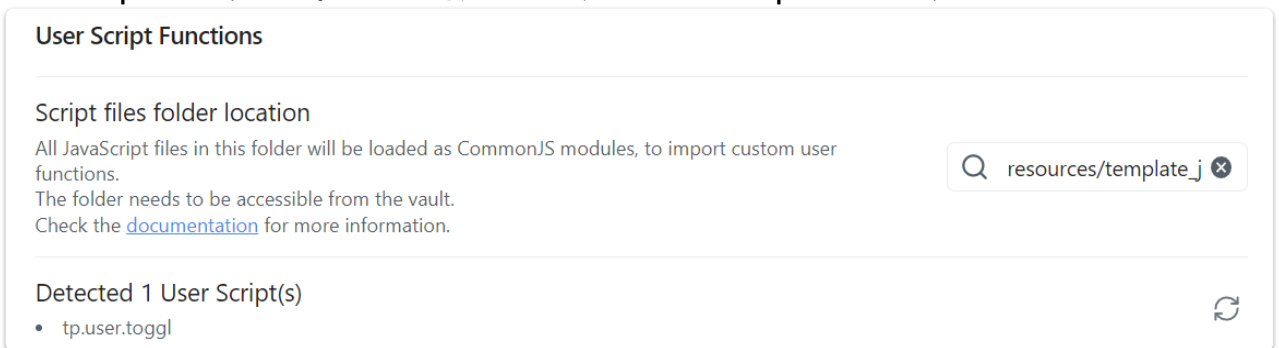


配置步骤

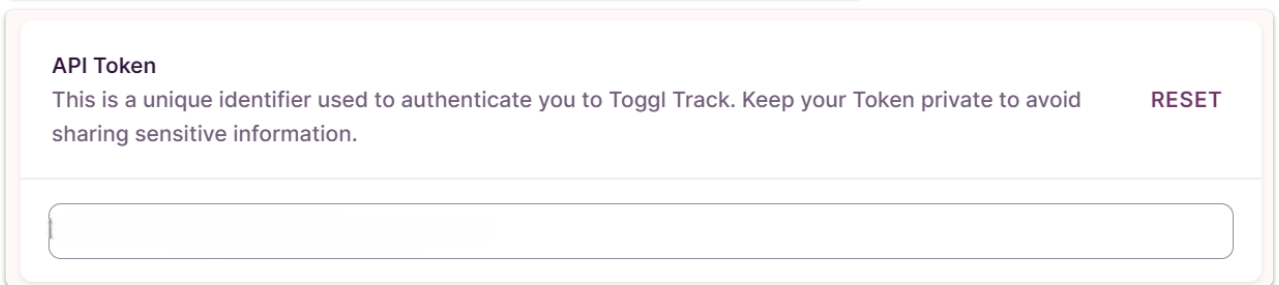
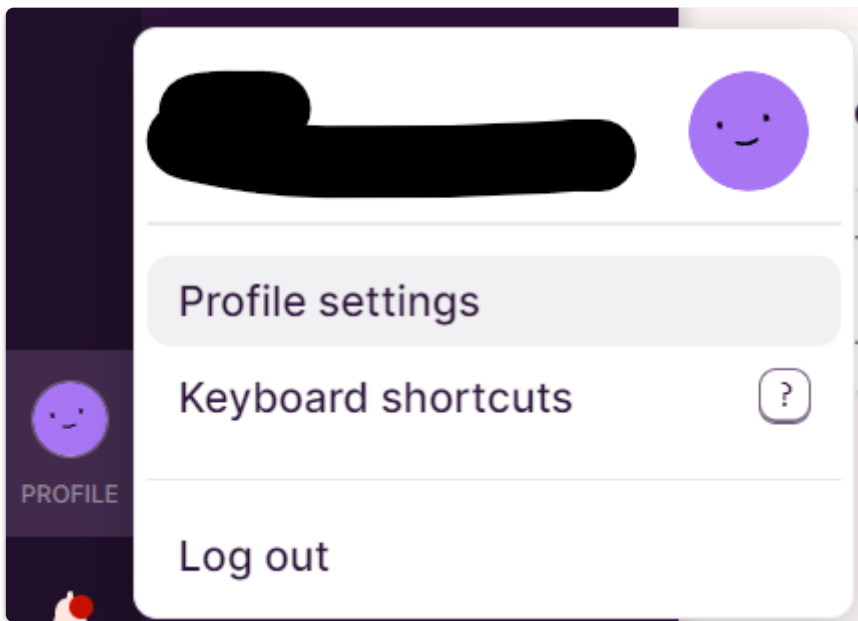
Templater配置

这里默认大家会最基本的templater使用，不会的话请参考[templater-obsidian](https://templater-obsidian.com/)

1. 把 `toggl.js` 放在任意obsidian目录下 (如 `resources/template_js/toggl.js`)
 1. 懂编程的小伙伴可以自行修改
2. 在Templater设置中指定对应目录为User Script的文件夹



3. 打开Toggl Profile Setting页，找到API Token。如果懒得找的话也可以使用 `email:password` 代替。



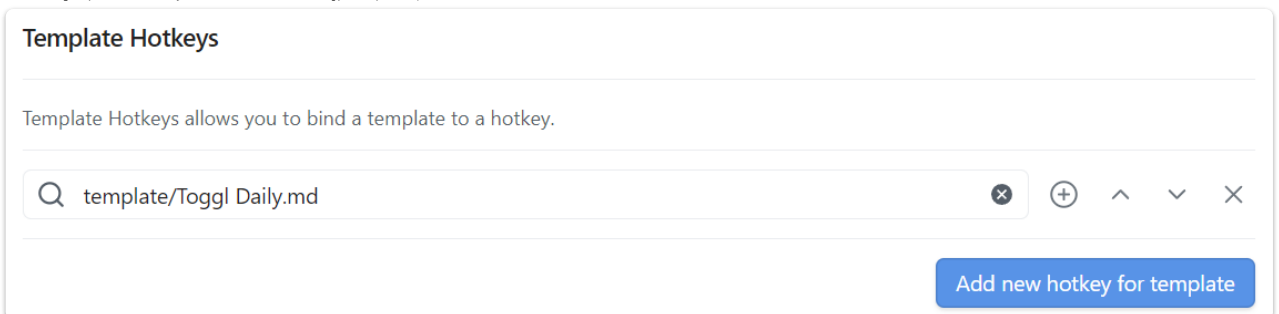
4. 创建新的template (参考Toggl Daily.md) , 这里需要把 `YOUR_API_TOKEN` 替换成刚刚找到的API Token。或者用 `email:password` 来代替

```
YOUR_API_TOKEN:api_token
```

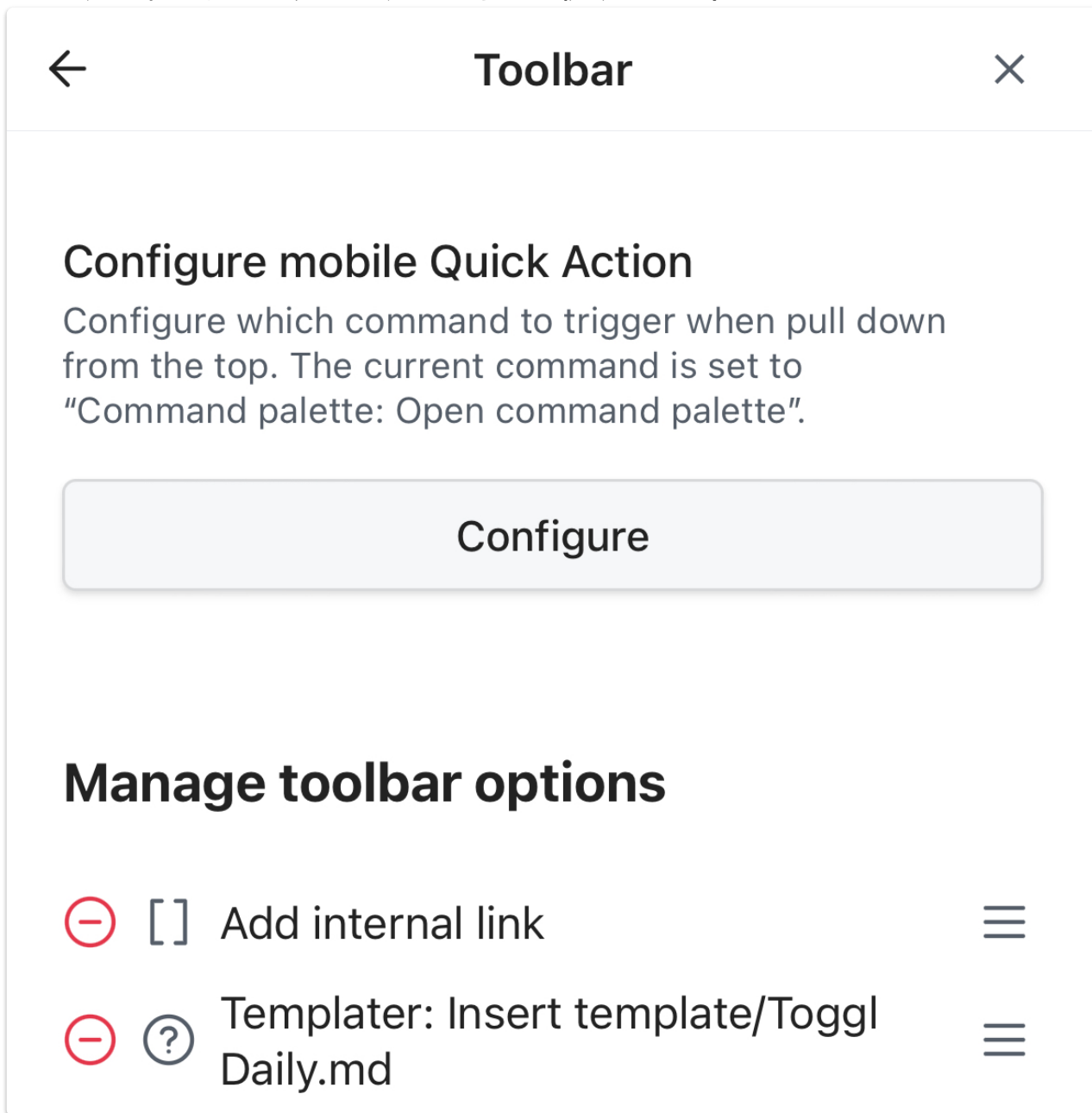
Toggl Daily

```
> [!info]- Time Log <% tp.date.now("kk:mm") %>
<% tp.user.toggl(tp, api token="YOUR_API_TOKEN:api_token") %>
```

5. 这个时候应该已经设置完成了, 为了使用方便, 我们可以在Templater插件中为它设置一个快捷键。



6. 在手机上，我们可以把它加入编辑栏快捷按钮中。



设置结束之后，就可以在光标处添加时间记录了。因为我后续会在 Tracker 插件中根据信息生成图表，所以会插在每日笔记下面。

Tracker 配置

每周时间花费饼图

下面是示例的代码块，但是具体怎么写还是需要参照自己的项目有哪些。效果已经在上面展示了。因为我是加在周记模板里的，所以 `startDate` 和 `endDate` 是自动填入的，具体就不赘述了，可以参考别的 Templater 教程。

```
```tracker
searchType: dvField
searchTarget: 学习, 工作, 休息, 运动, 日常, 社交, 睡觉, 其它
folder: daily
datasetName: 学习, 工作, 休息, 运动, 日常, 社交, 睡觉, 其它
startDate: 2024-04-28
endDate: 2024-05-05
pie:
 title: 每周时间花费
 data: '{{sum(dataset(0))}},{{sum(dataset(1))}},
{{sum(dataset(2))}},{{sum(dataset(3))}},{{sum(dataset(4))}},
{{sum(dataset(5))}},{{sum(dataset(6))}},{{sum(dataset(7))}},
{{{(24*7-sum(dataset(0))-sum(dataset(1))-sum(dataset(2))-
sum(dataset(3))-sum(dataset(4))-sum(dataset(5))-sum(dataset(6))-
sum(dataset(7))))}}}'
 dataColor: "#c7af14, #e36a00, #465bb3, #d94182, #06a893,
#990099, #0b83d9, #525266, #aaaaaa"
 label: '学习 {{sum(dataset(0))/24/7*100}}%, 工作
{{sum(dataset(1))/24/7*100}}%, 休息 {{sum(dataset(2))/24/7*100}}%,
运动 {{sum(dataset(3))/24/7*100}}%, 日常
{{sum(dataset(4))/24/7*100}}%, 社交 {{sum(dataset(5))/24/7*100}}%,
睡觉 {{sum(dataset(6))/24/7*100}}%, 其它
{{sum(dataset(7))/24/7*100}}%, 未记录 {{{(24*7-sum(dataset(0))-
sum(dataset(1))-sum(dataset(2))-sum(dataset(3))-sum(dataset(4))-
sum(dataset(5))-sum(dataset(6))-sum(dataset(7)))/24/7*100}}}'
 extLabel: '学习 {{sum(dataset(0))/24/7*100}}%, 工作
{{sum(dataset(1))/24/7*100}}%, 休息 {{sum(dataset(2))/24/7*100}}%,
运动 {{sum(dataset(3))/24/7*100}}%, 日常
{{sum(dataset(4))/24/7*100}}%, 社交 {{sum(dataset(5))/24/7*100}}%,
睡觉 {{sum(dataset(6))/24/7*100}}%, 其它
{{sum(dataset(7))/24/7*100}}%, 未记录 {{{(24*7-sum(dataset(0))-
sum(dataset(1))-sum(dataset(2))-sum(dataset(3))-sum(dataset(4))-
```

```
sum(dataset(5))-sum(dataset(6))-sum(dataset(7)))/24/7*100}}%'
 showExtLabelOnlyIfNoLabel: true
 ratioInnerRadius: 0.3
 ...
```

## 季度时间花费分布图

下面是示例的代码块，但是具体怎么写还是需要参照自己的项目有哪些。效果已经在上面展示了。

```
```tracker
searchType: dvField
searchTarget: 学习, 工作, 休息, 运动, 日常, 社交, 睡觉, 其它
folder: daily
startDate: -3M
penalty: 0
fitPanelWidth: true
datasetName: 学习, 工作, 休息, 运动, 日常, 社交, 睡觉, 其它
aspectRatio: 20:6.5
stack: true
bar:
  yAxisLabel: hr
  xAxisLabel: ''
  yMax: 24.1
  yMin: 0
  title: ''
  xAxisTickInterval: 1w
  yAxisTickInterval: 4
  barColor: "#c7af14, #e36a00, #465bb3, #d94182, #06a893,
#990099, #0b83d9, #525266"
  xAxisTickLabelFormat: MM-[W]WW
  showLegend: true
  ...
```

附录

Toggl Daily.md 完整代码

```
> [!info]- Time Log <% tp.date.now("kk:mm") %>
<% tp.user.toggl(tp, api_token="YOUR_API_TOKEN:api_token",
workspace_id=YOUR_WORKSPACE_ID) %>
```

如果不知道 workspace id的话, 也可以用下面的代码

```
> [!info]- Time Log <% tp.date.now("kk:mm") %>
<% tp.user.toggl(tp, api_token="YOUR_API_TOKEN:api_token") %>
```

如果不知道API_TOKEN的话, 也可以用下面的代码

```
> [!info]- Time Log <% tp.date.now("kk:mm") %>
<% tp.user.toggl(tp, api_token="email:password") %>
```

toggl.js 完整代码

```
async function toggl(tp, api_token, workspace_id=null) {

  Number.prototype.toHHMM = function () {
    var sec_num = this; // don't forget the second param
    var hours   = Math.floor(sec_num / 3600);
    var minutes = Math.floor((sec_num - (hours * 3600)) / 60);

    if (hours > 0){
      if (minutes == 0) {return hours+'hr'}
      return hours+'hr'+minutes;
    }
    return minutes+'min'
  };
};
```



```
// If workspace id is not provided, try to fetch it
if (workspace_id == null) {
    workspace_resp_async = requestUrl(
        {
            url:
'https://api.track.toggl.com/api/v9/workspaces',
            method: "GET",
            headers: {
                'content-type': 'application/json',
                "Authorization": "Basic " + btoa(api_token),
            }
        }
    );
    workspace_resp = await time_entries_resp_async;
    workspace = time_entries_resp.json;
    if (workspace.length > 0) {
        workspace_id = workspace[0].workspace_id;
    } else {
        return "Failed to fetch workspace";
    }
}

time_entries_resp_async = requestUrl(
    {
        url:
'https://api.track.toggl.com/api/v9/me/time_entries',
        method: "GET",
        headers: {
            'content-type': 'application/json',
            "Authorization": "Basic " + btoa(api_token),
        }
    }
);
projects_async = await requestUrl(
    {
        url: 'https://api.track.toggl.com/api/v9/workspaces/' +
workspace_id + '/projects',
```

```

        method: "GET",
        headers: {
            'content-type': 'application/json',
            "Authorization": "Basic " + btoa(api_token),
        }
    }
);
time_entries_resp = await time_entries_resp_async;
time_entries = time_entries_resp.json;
projects = (await projects_async).json;
project_map = {}
projects.forEach((item) => {
    project_map[item.id] = item.name
});
sum_map = {}
time_entries.forEach((item) => {
    if (item.stop && Date.parse(item.stop) >= (Date.now() -
24*3600*1000)) {
        project_name = project_map[item.project_id];
        if (!sum_map.hasOwnProperty(project_name)) {
            sum_map[project_name] = {}
        };
        key = [item.tags.join(','),
item.description].filter(function (el) {
            return el != null && el != '';
        }).join('-');
        if (!sum_map[project_name].hasOwnProperty(key)) {
            sum_map[project_name][key] = 0
        };
        sum_map[project_name][key] += item.duration;
    }
});

result = "";
for (var project in sum_map) {
    var tasks = [];
    var total_tasks = Object.keys(sum_map[project]).length;

```

```

        var total_time = Object.values(sum_map[project]).reduce((a,
b) => a + b, 0);
        result += project + ':: ' + (total_time / 3600).toFixed(1)
+ '\n';
    }
    result += '\n';
    for (var project in sum_map) {
        var tasks = [];
        var total_tasks = Object.keys(sum_map[project]).length;
        for (var task in sum_map[project]) {
            tasks.push(((total_tasks == 1 && task == project)?"" :
task + ": ") + sum_map[project][task].toHHMM())
        };
        task_str = tasks.join("; ");
        result += ' [' + project + '] --' + task_str + ".";
        if (total_tasks > 1) {
            result += " Total: " +
Object.values(sum_map[project]).reduce((a, b) => a + b, 0).toHHMM()
+ "."
        };
        result += "\n";
    }
    return result;
}

```

```

module.exports = toggl

```