# tournamenter Documentation

*Release 0.0.1*

**Ivan Seidel and RoboCupDevs**

May 20, 2014

# Contents

*the last tool you need to manage your tournament*

The system called “tournamenter” was made specificly for the RoboCup 2014, with the intention of working locally, and serving as a helper for storing and sharing real-time information about scores, rounds, soccer matches and most of what spectators and teams need to know.

One of the objectives was to allow others (even a small competition), to easily use it. The second objective was to make it useful for most of the tournament types, currently in RoboCup (Rescue, Soccer, Dance, @Home…), and also outside RoboCup (FLL, Sumo…).

For every tournament one system is required. We didn’t want to mix things up. During RoboCup, about 20 systems like this will be running simultaneously to provide access to Chairs for each league.

Contents:

CHAPTER 1

# Introduction

Before we start talking about the system there are a few names that are good to know. Here is a collection of helper names with their meaning:

- **View**: Something that will be shown to spectators. It has a group of Pages. Like an presentation
- **Page**: A single slide on a View. Imagine it as a slide of a presentation
- **Page type**: There are many page-types, such as: Table View, CSV View, Message View, Group View...
- **Group**: A group is composed of soccer matches. A group also generates a scoring table in soccer style
- **Table**: A table representing Teams in it's row, and scores in its columns.
- **Score**: A single entry of points for a team
- **Final Score:** Computed with all its Scores. You can define a custom function to compute it.
- **Module**: A part of the system.

## 1.1 Modules

We divided the system in four main modules:

- Teams
- Groups
- Tables
- Views

# Installation

## 2.1 Linux/Mac or other UNIX based OS

### 2.1.1 *nodejs* installation

Since *tournamenter* is a NodeJS based web app you need to have *nodejs* and *npm* installed on your machine.

For instance on Ubuntu/Debian you can run:

```
$ sudo apt-get install nodejs
```

On Mac with *homebrew* you can install it by running:

```
$ brew install node
```

### 2.1.2 Downloading and installing tournamenter from sources

First of all, you need to get the latest version of the *tournamenter* source code repository. You can do so by typing this command to the command line::

```
$ git clone --depth 1 https://github.com/RoboCupDev/tournamenter.git
```

After that, you should have a *tournamenter* directory in your current working directory. Since we already have *nodejs* and *npm* installed we can navigate into the *tournamenter/* directory and run:

```
$ npm install
```

Which will install the required packages for tournamenter to work correctly. In order to see if everything works as expected you can type:

```
$ npm start
```

Which will make the server start at http://127.0.0.1:1337 and with a bit of luck you should see your own *tournamenter* instance there!

### 2.1.3 If you are out of luck ...

Although it should not it might happen that the command above will finish with error message. If the error looks like:

```
TypeError: Cannot call method 'createCollection' of undefined
```

then you can easily fix the situation by starting *tournamenter* by running:

```
$ node app --adapter.module "sails-disk"
```

which will use your hard drive to save the data you create while playing with tournamenter. Again, you can verify that everything is working as expected by looking at http://127.0.0.1:1337