# Development testing

This chapter will focus on the implementation of the test application described in chapter (TODO: crossref testapp methodology). Each framework has its own subsection where I describe every aspect of the development and thoughts I made during the process. Each section has its own summary, and finally, there is a conclusion giving a nuanced look at what framework solves the different tasks of a real-time application best.

## Socket.IO

Socket.io is a module for Node.js (TODO: nodejs) that provides real-time through pure JavaScript on both server and client. It has been around since ca. 2011 (TODO: last commitpage)[[1]](#footnote-1), and it aims to provide clean and simple real-time across all platforms:

*“Socket.IOaims to make realtime apps possible in every browser and mobile device, blurring the difference between the different transport mechanisms”.*

While it hasn’t reached 1.0 yet, it is used in production code by several companies, and it is considered stable. Perhaps one of the most “famous” applications that use Socket.IO is Trello – the online “Scrum Board”[[2]](#footnote-2).

### Why I chose it

Node.js is increasingly popular, and the idea of using JavaScript on the server is very exciting! Over the past couple of years, there has been a dramatic change in the way developers think of JavaScript (TODO: sources). Therefore, it was only natural that I chose at least one framework that uses Node.js as server.

Though there are several modules for Node that provides real-time (TODO: link to node modules), Socket.IO stands out from the crowd.

## Lightstreamer

## Play Framework

## SignalR

## Meteor

## Conclusion

1. 0.7 preview was released May 5th 2011 [↑](#footnote-ref-1)
2. <https://trello.com/> [↑](#footnote-ref-2)