The Journal of Open Source Software

CRSocket: A web app component facilitating the administration of digital trials from a separate device

Henrik Dvergsdal[1]

1 Nord University, Norway

DOI: 10.21105/joss.05658

Software

- Review
- Repository
- Archive

Editor: Britta Westner

Reviewers:

- @rabdill
- @lukaszjablonski

Submitted: 15 May 2023
Published: 31 August 2023

Summary

The Controller-Responder Socket (`CRSocket`) is a web app component that can be used to develop digital tasks where testers use a controller device to administer trials and a separate responder device to present stimuli and collect responses. In addition to providing a more streamlined and comfortable test situation, this approach may improve measurement quality because potentially distracting activities related to configuration and initiation of tasks are hidden from the participant. It also facilitates more flexible data collection protocols where testers can make real-time adjustments based on direct observations of the test situation and feedback from various sources presented on the controller device. In addition to communication functionality, the `CRSocket` also provides client-side buffering of events and server-side plug-in interfaces related to access control and data storage. Please note that usage of this software requires programming skills and experience with the PHP and JavaScript programming languages.

Statement of need

The `CRSocket` component was originally developed for the Early Childhood Inhibitory Touch Task (ECITT). This is a suite of experimental psychology tasks designed to measure inhibitory behaviour using trials presented on tablet computers.

The primary novelty of the ECITT is that it can be "minimally modified to suit different ages, whilst remaining structurally equivalent" (Holmboe et al., 2021). This is the first task that can be used to measure inhibitory behaviour in a consistent way across the lifespan, from 18 months and upward. In addition to the longitudinal validation publication, three studies related to specific age groups based on this task, have been published so far (Fiske et al., 2022; Hendry et al., 2022; Lui et al., 2021).

Tablet computers and similar devices have become a natural part of daily life and are generally considered feasible for data collection within various humanities disciplines (Frank et al., 2016; Kanerva et al., 2019; Semmelmann et al., 2016), but the abstract nature of such tasks still require careful consideration and adaption to individual differences among participants and the context in which data is collected (Wenz, 2021). Using tablet tasks in developmental psychology research is challenging because the capacity to engage with stimuli presented on two-dimensional touch screens varies with age. Pre-verbal infants pose a challenge in most types of experimental research because they have to be introduced to tasks by means of social interaction and hands-on experiences rather than plain instructions. On tablet devices, this will typically involve different types of games and priming trials. Since infants are generally more distractible and impulsive than other age groups, it is particularly important to minimise irrelevant incidents and present tasks at the right moments. Jenkins et al. (2016) has identified similar challenges with older respondents. With this age group, one also has to take physical conditions such as visual acuity and motor skills into account.

To my knowledge, there is currently no other software designed for digital tasks that implement controller-responder functionality. Hence, current approaches to tablet tasks are either based on self-administration or on handing over pre-configured devices to the participant. Controlling tasks from a separate device can help meet some of the challenges mentioned above by enabling a bigger toolbox related to motivation, preparation and individual task adaption, based on researchers' ability to engage with and understand the participants.

Although the component was originally designed for lab experiments, it may also contribute to improved measurement quality and the creation of novel tablet tasks for other purposes, including settings where testers are not located in the same room but are able to communicate in other ways, e.g. via video channels.

Architecture

To maximise reusability, the component has been extracted from the original code (an undocumented prototype included in Holmboe et al. (2021) as supplementary material) and encapsulated in generic classes, using only built-in client- and server-side functionality and standard APIs. There are no global variables and no bindings to other parts of the ECITT software or external libraries. Client-side classes are implemented in JavaScript and server-side classes are implemented in PHP.

The component implements a relatively robust protocol for the identification and exchange of events between controller and responder processes. It is designed to handle various technical and human incidents that have been observed during testing. This includes differences between different browser engines, network issues, and challenges related to server functionality, resource usage, and delays.

The protocol is asymmetric in the sense that only controller apps can establish and maintain communication channels. Responder apps can accept, reject, and abort connections, but they cannot actively approach controllers to establish new connections. Apart from that, controllers and responders interact as equal peers.

The `CRSocket` class

The `CRSocket` class is used by controller and responder client apps to establish communication channels, exchange `CREvent` objects, and monitor communication states. `CRSocket` objects are instantiated with a client role which can be either "controller" or "responder" and a `CRConfig` object which, among other things, contains an application ID that identifies a controller/responder web app pair.

The first time a `CRSocket` object is instantiated, it is assigned a server-generated client ID. This ID is associated with the current combination of application ID / client role within the current engine. It is saved in the engine's `localStorage` object and can be reused in subsequent instantiations to establish communication channels between active `CRSocket` objects in all major browser engines.

The class is implemented as a state machine, with a well-defined set of states and state transitions. Event reception and state changes are communicated by means of standard JavaScript event APIs.

The `CREvent` class

This class, with mirroring implementations in JavaScript and PHP, represents events that can be transmitted between controller and responder `CRSocket` objects. These events may contain any number of text-convertible parameters. They also contain flags related to buffering, logging, and server-side interception.

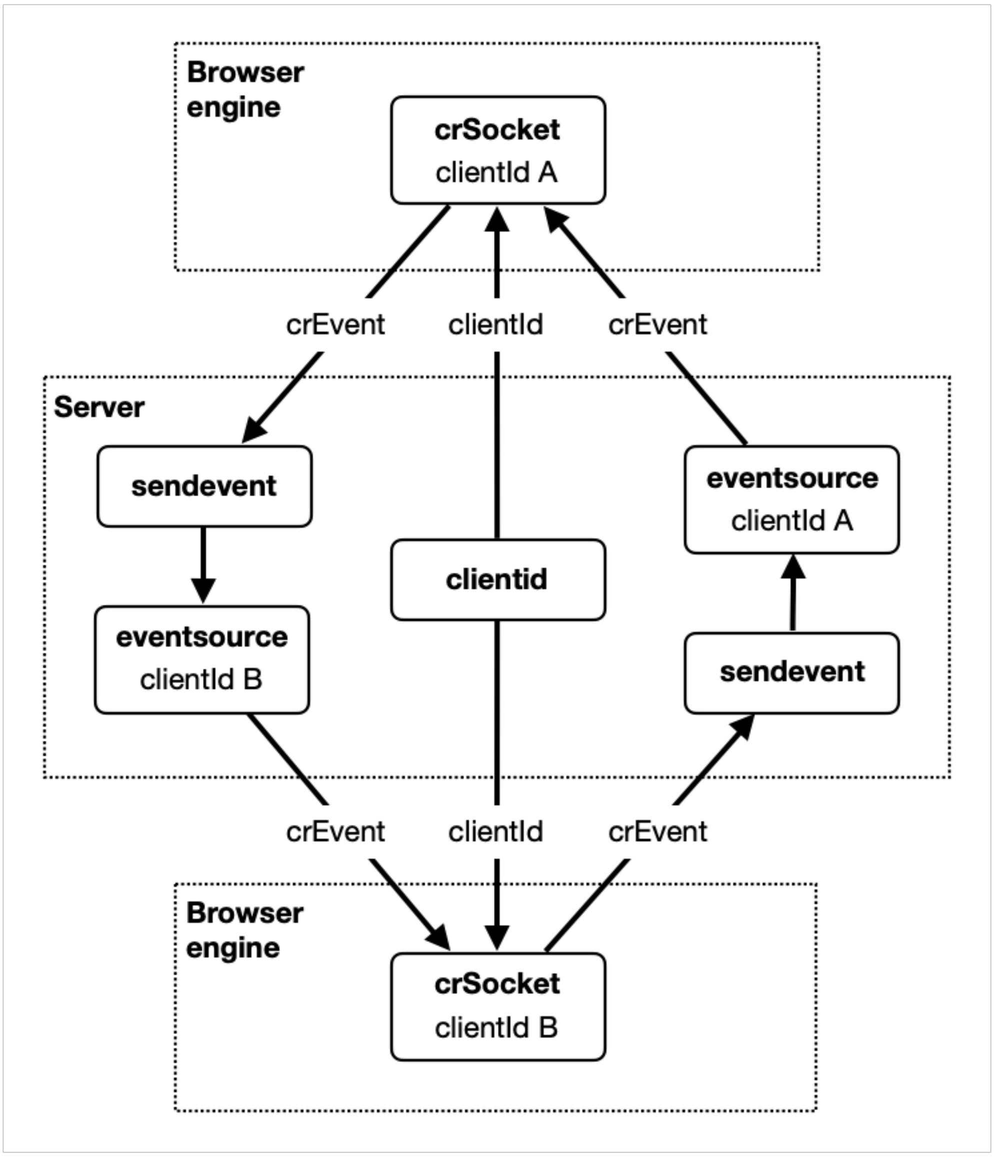
Server resources

The component provides default implementations of resources that must be available on the server:

- **`clientid`**. This resource generates a new, unique `clientId`, and returns an XML document containing this ID to the requesting client.
- **`eventsource`**. This resource transmits a stream of events, representing `CREvent` objects and internal events from the server to the client, formatted as server-sent events.
- **`sendevent`**. This resource receives a single event from a client, formatted as an HTTP GET request and transmits it to an eventsource process.

Server-side functionality can be replaced by other implementations as long as the resources described above conform to their respective specifications.

Both `eventsource` and `sendevent` resources may contain server-side event plugin mechanisms that can be used to implement authentication schemes, storage functionality and custom event processing. User authentication is based on access tokens. This facilitates integration with a wide range of authentication schemes ranging from simple user ids to advanced industry-standard protocols.



Basic architecture. Illustrates how `CRSocket` objects interact with server resources to obtain unique identifiers and exchange `CREvent` objects.

Naming conventions:

- Classes and data types are named with medial capitals, starting with capital letters, e.g. `CRSocket`.
- Class instances (objects) and variables are named with medial capitals, starting with lowercase letters, e.g. `crSocket`.
- Server resource are named with lowercase letters only, e.g. `eventsource`.

Acknowledgements

I would like to thank Andrew Simpson who came up with the original idea and Karla Holmboe who brought it into light by contributing with methodological expertise and integration into several developmental psychology research projects. I also appreciate the work of all the testers who provided useful feedback while patiently dealing with bugs and deficiencies.

References

Fiske, A., Klerk, C. de, Lui, K. Y. K., Collins-Jones, L., Hendry, A., Greenhalgh, I., Hall, A., Scerif, G., Dvergsdal, H., & Holmboe, K. (2022). The neural correlates of inhibitory control in 10-month-old infants: A functional near-infrared spectroscopy study. *NeuroImage*, *257*. https://doi.org/10.1016/j.neuroimage.2022.119241

Frank, M. C., Sugarman, E., Horowitz, A. C., Lewis, M. L., & Yurovsky, D. (2016). Using tablets to collect data from young children. *Journal of Cognition and Development*, *17*. https://doi.org/10.1080/15248372.2015.1061528

Hendry, A., Greenhalgh, I., Bailey, R., Fiske, A., Dvergsdal, H., & Holmboe, K. (2022). Development of directed global inhibition, competitive inhibition and behavioural inhibition during the transition between infancy and toddlerhood. *Developmental Science*, *25*. https://doi.org/10.1111/desc.13193

Holmboe, K., Larkman, C., Klerk, C. de, Simpson, A., Bell, M. A., Patton, L., Christodoulou, C., & Dvergsdal, H. (2021). The early childhood inhibitory touchscreen task: A new measure of response inhibition in toddlerhood and across the lifespan. *Plos One*, *16*. https://doi.org/10.1371/journal.pone.0260695

Jenkins, A., Lindsay, S., Eslambolchilar, P., Thornton, I., & Tales, A. (2016). Administering cognitive tests through touch screen tablet devices: Potential issues. *Journal of Alzheimer's Disease*, *54*. https://doi.org/10.3233/JAD-160545

Kanerva, K., Kiistala, I., Kalakoski, V., Hirvonen, R., Ahonen, T., & Kiuru, N. (2019). The feasibility of working memory tablet tasks in predicting scholastic skills in classroom settings. *Applied Cognitive Psychology*, *33*. https://doi.org/10.1002/acp.3569

Lui, K. Y. K., Hendry, A., Fiske, A., Dvergsdal, H., & Holmboe, K. (2021). Associations between touchscreen exposure and hot and cool inhibitory control in 10-month-old infants. *Infant Behaviour & Development*, *65*. https://doi.org/10.1016/j.infbeh.2021.101649

Semmelmann, K., Nordt, M., Sommer, K., Röhnke, R., Mount, L., Prüfer, H., Terwiel, S., Meissner, T. W., Koldewyn, K., & Weigelt, S. (2016). U can touch this: How tablets can be used to study cognitive development. *Frontiers in Psychology*, *7*. https://doi.org/10.3389/fpsyg.2016.01021

Wenz, A. (2021). Do distractions during web survey completion affect data quality? Findings from a laboratory experiment. *Social Science Computer Review*, *39*. https://doi.org/10.1177/0894439319851503