Teddy App Documentation

Introduction

Teddy is a research platform that records, encrypts, and uploads facial recordings of users as they watch viral videos and talk about their past 24 hours. It is available both for Android and iOS.

Teddy can be downloaded at https://teddy.media.mit.edu/

Furthermore, Teddy's privacy policy is available at https://teddy.media.mit.edu/privacy/

Testing

To access Teddy, you need a username and password. If you make an account with the word "test" in it, you can make up any password you want, and it will let you in. Please contact us if you'd like more specific test accounts created, or if you want to create accounts for your study participants to be able to use without deeplinks.

Testing Using Deep Links

You can easily test the functionality of opening Teddy with a deep link. Simply install Teddy on your phone, then on that same phone go to this website:

https://teddy.media.mit.edu/test/

When you click on any of the links there, it will open Teddy and automatically log you in with the appropriate account info depending on which link you click. Feel free to try them all out!

Deep Link

Teddy can (and should!) be accessed by your study participants via a deep link. Create a link at the end of a Redhat or Qualtrix form, and it will open the Teddy app if they have it installed. Teddy's Deep Link functionality is very complex, so please read carefully.

The deep link format is as follows:

```
affectteddy://projectname/participantid/(sessionid?)/(sessionType?)/(nextURL?)
```

The parameters are as follows:

`projectname (required)`
The project name for your study. We will assign you a project name you should use for your actual study, and a project name you can use for testing. These project names are used in our Data API (see below) so do not make one up without checking with us, please.

`participantid (required)`
The participant ID you want for this participant. Do not use their email, real name, or any identifiable information. It is best if this is a short random sequence of letters and numbers.

`sessionid (optional, unless sessionType is also included)`
This is an ID you can use to track this particular App Session. An App Session is defined as a single experience of recording movies within Teddy. The session ID is not required (unless you include a sessionType as well) and can be useful for you to track which app session is linked with which answers in Qualtrix/Redhat. Session IDs should be unique within your project.

`sessionType (optional - default: “all”)`
SessionType is an optional argument and can be three values: `all` or `video` or `story`.

- `video -` The app only performs the first recording task, asking the user to watch a viral video.
- `story -` The app only performs the second recording task, asking the user to talk about their past 24 hours.
- `all -` The app performs both recording task, with the viral video first, then talking about the past 24 hours

`nextURL (optional)`
The next URL is the URL that Teddy should open once the current app session is completed. It is used in case you want the user to continue filling out more surveys after they use the app. Currently, for security purposes, the nextURL **must** start with `https://redcap.ictr.wisc.edu/`. If you need the nextURL parameter to go to any other domain, we need to edit the app to allow said domain. The nextURL parameter can be a raw URL, or a URL-encoded string like `https%3A%2F%2Fredcap.ictr.wisc.edu%2F` The app does its best to handle both forms of URLs.

Projects

The backend defines the concept of a Project. A project is defined by its Project Name and contains several attributes.

- Name - The name of the project used to identify it in deep links and the data API.
- Description - This is never exposed, but is useful for keeping notes on Projects in the Django admin.

- Video Set - The video set this project should use. See the “Current Teddy Viral Videos” section for values and meanings.
- Randomize Videos - If set to true, each user will shuffle their list of videos upon first entering the app. Note that modifying this field will only affect future users, not any users who have already begun using the app.

Data API

You can work with our team to register a Data Access Account, which can be used to query app sessions from our database. This is really useful to make sure your users are completing their sessions on time. Please contact us if you do not have a Data Access Account and would like one, and we’ll create one for you.

Do note that you cannot download videos automatically. We need to send them to you manually, for security reasons. This API only returns the meta-data around a session such as the viral video shown, start and end time, and session ID as sent via your deep links.

Once you have a Data Access Account, you can access it via the API endpoint we have created. Sorry, there is no user-friendly webpage for it, yet. We can make one if you’d like.

To access the endpoint, use the following command:

```
curl -u username:password https://teddy.media.mit.edu/querysessions/
```

Note that it takes several optional arguments to filter the results. Hopefully they're self explanatory, but you can use any combination you want:

- `session_label`
    - (this is the `sessionid` from the deeplink. Sorry it has a different name here, I’ll fix that later.)
- `user_id`
    - (this is the `participantid` from the deeplink. Ditto about the confusing naming.)
- `project_name`
    - (this is the `projectname` from the deep link and should be one of your Project’s names. If left out, it will show data for all Projects that your Data Access Account has been cleared to access. If you’ve been using a projectname that your Account hasn’t been cleared to access, please contact us and we’ll get you squared away.)
- `Date_range_start`
    - (only looks at the session start date of the app sessions, any reasonable date format should work)
- `Date_range_end`

    - (likewise, filters based on the end date of the app session.)

For example, you can query

```
https://teddy.media.mit.edu/querysessions/?userid=tim&date_range_start=2001-01-01
```

And that will return any sessions by user “tim” after January 1st 2001.

Technology

Teddy was created using React Native, and the backend was created in Django. All videos are encrypted before being uploaded using the VirgilCrypt library to encode the videos using AES-256 encryption, whose key is then encrypted with an RSA-2048 public key, the private key of which is never uploaded to our backend servers.

Current Teddy Viral Videos

Videos are rotated through and shown in the same order for everyone, unless the Project in the backend has been set to randomize the video order.

Individual projects can be set to view specific sets of videos. Currently there are two sets of videos, and an “all” set that includes every video. The default value is “babies.”

Video Set “babies”

This set was developed by the folks at UW-Madison based on their research of trying to elicit smiles.

- `babygreen`
    - https://www.youtube.com/watch?v=GGzgX2XJ9nw
    - “What to Expect When Baby Laughs”, 6 thousand views, 0:55
- `babysneeze`
    - https://www.youtube.com/watch?v=VXRA7KGN5pg
    - “baby iQi laughing when mother sneezing”, 90 views, 1:00
- `babycomp1`
    - https://youtu.be/TDPP1F4rD40?t=42
    - “BABY LAUGHS ARE THE BEST LAUGHS”, 8 views, 1:00 (0:42-1:42)
- `babycomp2`
    - https://youtu.be/TDPP1F4rD40?t=42
    - “BABY LAUGHS ARE THE BEST LAUGHS”, 8 views, 1:00 (2:00-3:00)
- `babydonkey`
    - https://www.youtube.com/watch?v=BOpNAHk0wZI
    - “Baby Laughs When Mom Says "Donkey"”, 3 million views, 0:54

Video Set “misc”

These videos were all picked for because they have no previous required knowledge, and they are not mean spirited. My personal favorites are calmowl and frogsqueak.

- jackrussell
    - https://www.youtube.com/watch?v=A4N7G29GWQI
    - “Hilarious Jack Russell Goes Crazy with Excitement at Crufts 2017!”, 15 million views, 1:15
- lawyercat
    - https://www.youtube.com/watch?v=TDNP-SWgn2w
    - “Filter turns lawyer into cat during Zoom hearing”, 1.6 million views, 0:39
- babylaugh
    - https://www.youtube.com/watch?v=RP4abiHdQpc&ab_channel=BruBearBaby
    - “Baby Laughing Hysterically at Ripping Paper”, 20 million views, 1:00
- calmowl
    - https://www.youtube.com/watch?v=FBgLytbB-uE
    - “Calm Owl”, 6 million views, 0:05
- frogsqueak
    - https://www.youtube.com/watch?v=HBxn56l9WcU
    - “A tiny angry squeaking Frog 🐸”, 17 million views, 0:45
- hellodarkness
    - https://www.youtube.com/watch?v=02-QOC9hNO4
    - Existential Guineaupig, 1.5 million views, 0:42

Video Set “all”

This set includes all videos uploaded into the app, in the order presented here.