

# Stealing Ur Feelings Project Documentation

## Overview

"Stealing Ur Feelings" emerged from a complex development process, marked by trial, error, and intense programming sprints. This document aims to outline the project's evolution, technical challenges, and innovative solutions employed to bring it to life. Initially premiered at the Tribeca Film Festival in April 2019, the project underwent significant optimization and development phases to achieve its current state.

## Technical Challenges and Solutions

### Computer Vision Engines

Throughout its development, "Stealing Ur Feelings" employed various computer vision engines to power its emotion recognition capabilities. The project's approach evolved from a complex custom engine based on TensorFlow to more streamlined solutions, including:

- A scrapped TensorFlow-based custom engine.
- Dlib compiled for the browser via WebAssembly, leveraging multithreaded facial landmark detection.
- The adoption of face-api.js in the final version for enhanced performance and reliability.

## Video Synchronization

A novel system was developed for frame-accurate video synchronization, utilizing encoded information embedded in the video's overscan area. This method, inspired by datacasting, required precise arithmetic solutions to overcome floating-point approximation errors.

## Browser-Specific Enhancements

Significant effort was made to optimize the project for different web browsers:

- Chrome: Implementation of a zero-copy hack using OffscreenCanvas to execute the computer vision engine on a separate thread, resulting in smoother performance.
- Firefox: Challenges arose due to incomplete OffscreenCanvas implementation, impacting performance.

## Unique Features

### Secret Physical Installation Mode

A highlight of the project is its secret physical installation mode, debuted at the Tribeca Film Festival. This mode involved:

- A large television display hanging from the ceiling, revealing the AI-extracted "secrets" of participants in real-time.
- A specialized client-server setup located in the `/installation` directory.
- For the premiere, a humorous twist was added by adjusting the political bias algorithm to classify all participants as far-right conservative Republicans, leading to memorable reactions.

## Preservation of Development Artifacts

In a nod to the project's tumultuous development journey, all original comments, including those difficult to decipher, have been preserved in the codebase. This decision serves as a testament to the challenges faced and the iterative nature of the project's evolution.

# Service Package

This document provides an overview and detailed breakdown of the `service` package, version `1.0.0`. The package is a comprehensive collection of dependencies crucial for the functioning of a service-oriented application. Below is an analysis of its structure, key dependencies, and their roles within the package.

## Overview

- Package Name: `service`
- Version: `1.0.0`
- Lockfile Version: `1` (Indicates the version of the lockfile format used)
- Requires: `true` (Specifies that each dependency listed in the lockfile includes the required dependencies)

## Dependencies

The `service` package includes a variety of dependencies, each serving a specific function. Here's a brief look at some of the key dependencies and their purposes:

### 1. `accepts` (version `1.3.5`)

- Purpose: Handles content negotiation for the service by determining the best match for the types offered by the server's responses.
- Dependencies: Requires `mime-types` (`~2.1.18`) for resolving MIME types and `negotiator` (`0.6.1`) for negotiating client/server preferences.

### 2. `after` (version `0.8.2`)

- Purpose: Used to execute asynchronous callbacks sequentially, ensuring that a particular function runs after a set number of other functions have completed.

### 3. `arraybuffer.slice` (version `0.0.7`)

- Purpose: Provides a method to slice an `ArrayBuffer`, similar to how `Array.prototype.slice` works, but for `ArrayBuffers`.

#### 4. `async-limiter` (version 1.0.0)

- Purpose: A tool for limiting the number of asynchronous operations running in parallel, useful for controlling resource utilization.

#### 5. `debug` (version 4.1.1)

- Purpose: A debugging utility that simplifies the process of debugging applications. It is modular and can be easily enabled or disabled.
- Dependencies: Requires `ms` (^2.1.1) to convert various time formats to milliseconds.

#### 6. `engine.io` and `engine.io-client` (version 3.3.2)

- Purpose: Provides the core engine for real-time communication between clients and servers.
- Dependencies: These packages depend on a number of modules like `ws` for **WebSocket support**, `debug` for debugging, and `cookie` for parsing and serializing cookie strings.

#### 7. `socket.io` and `socket.io-client` (version 2.2.0)

- Purpose: Enable real-time bidirectional event-based communication. They are built on top of `engine.io` and `engine.io-client` respectively, extending their capabilities.
- Dependencies: They share dependencies with `engine.io` and `engine.io-client`, including `debug` and `component-emitter` for event emitting.

#### 8. `mime-types` (version 2.1.22)

- Purpose: A library that allows for the lookup of MIME types based on extension and vice versa.
- Dependencies: Relies on `mime-db` (~1.38.0) for the database of MIME types.

**The `service` package version 1.0.0 is a comprehensive toolkit designed to facilitate the development of service-oriented applications with real-time capabilities. It leverages a wide array of dependencies to handle everything from content negotiation and MIME type resolution to real-time communication and asynchronous operations management. This document has outlined the key components and their roles within the package, providing a clear understanding of its functionality and utility.**

## **Conclusion**

"Stealing Ur Feelings" stands as a testament to the innovative, albeit chaotic, development process. Through its evolution, the project has explored the boundaries of emotion recognition technology, browser-specific optimizations, and the potential of interactive installations. Despite its challenges, the project has achieved a unique place in the intersection of technology, art, and social commentary.