

# readme.md for @start.plus/start.plus

An open-source startpage SaaS for families and companies.

## Install

To install the package:

```
npm install @start.plus/start.plus  
# or  
pnpm install @start.plus/start.plus
```

## Usage

### Running the Server

Start the development server:

```
pnpm start
```

This will serve the startpage application on `http://localhost:3000`.

### Development Mode

For development with live reload:

```
pnpm watch
```

## Building

To build the project:

```
pnpm build
```

## CLI Usage

The package provides a CLI command `startplus`:

```
startplus
```

## Features

### Dashboard

The main startpage view featuring:

- **Time-based greeting** - Dynamic greeting based on time of day (Good morning/afternoon/evening)
- **Current date display** - Full formatted date
- **Widget grid** - Responsive layout for productivity widgets
- **Search integration** - Quick access to search functionality

## Widgets

Built-in productivity widgets:

Widget	Description
<b>Clock</b>	Digital clock with date display and timezone support
<b>Search</b>	Multi-engine search bar with Google, DuckDuckGo, Bing, and custom search options
<b>Notes</b>	Quick markdown notes with localStorage persistence
<b>Bookmarks</b>	Quick access bookmark tiles with favicon support

## Bookmark Management

- Create, edit, and delete bookmarks

- Organize bookmarks into folders
- Automatic favicon fetching
- Search functionality
- Import/Export capabilities

# Application Shell

Built on `@design.estate/dees-catalog` with:

- Full application shell with routing
- Main menu navigation
- App bar with menus and user profile
- Keyboard shortcuts
- State persistence
- Dark theme

## Views

View	Description
Dashboard	Main startpage with widgets
Bookmarks	Bookmark management interface
Settings	Application settings

## Architecture

## Tech Stack

Category	Technology
Frontend	TypeScript, LitElement, Web Components
UI Library	<code>@design.estate/dees-catalog</code>
State	Custom reactive state with localStorage persistence
Backend	<code>@api.global/typedserver</code>
Build	<code>@git.zone/tsbuild</code> , <code>@git.zone/tsbundle</code>
Dev Server	<code>@git.zone/tswatch</code>

# Project Structure

```
start.plus/  
├─ ts/                # Backend  
│  ├─ index.ts       # Entry point  
│  ├─ plugins.ts     # Dependency imports  
│  ├─ paths.ts       # Path utilities  
│  └─ server/        # Server classes  
├─ ts_web/           # Frontend  
│  ├─ index.ts       # Entry point  
│  ├─ plugins.ts     # Dependency imports  
│  ├─ app/           # Main app component  
│  ├─ views/         # View components  
│  ├─ components/    # Widget components  
│  ├─ state/         # State management  
│  └─ interfaces/    # TypeScript interfaces  
├─ html/             # Static HTML files  
├─ dist_bundle/      # Built bundle output  
└─ test/             # Tests
```

## State Management

The application uses a simple reactive state system with localStorage persistence:

```
import { appState } from './state/state.js';  
  
// Get current state  
const bookmarks = appState.bookmarks.getState();  
  
// Update state  
appState.bookmarks.setState([...bookmarks, newBookmark]);  
  
// Subscribe to changes  
appState.bookmarks.select().subscribe((bookmarks) => {  
  console.log('Bookmarks updated:', bookmarks);  
});
```

# Configuration

## Environment Variables

Variable	Default	Description
PORT	3000	Server port
NODE_ENV	development	Environment mode

## Server Options

```
import { StartPlusServer } from '@start.plus/start.plus';

const server = new StartPlusServer({
  port: 3000,
  development: true,
});

await server.start();
```

# Development

## Prerequisites

- Node.js 18+
- pnpm

## Setup

```
# Clone the repository
git clone https://code.foss.global/start.plus/start.plus

# Install dependencies
```

```
pnpm install
```

```
# Start development
```

```
pnpm watch
```

## Testing

```
pnpm test
```

## Building

```
pnpm build
```

# PWA Support

Start.plus includes PWA (Progressive Web App) support with:

- Web app manifest
- Standalone display mode
- Theme color configuration
- Custom icon

## API

### StartPlusServer

```
class StartPlusServer {
  constructor(options?: IStartPlusServerOptions);
  start(): Promise<void>;
  stop(): Promise<void>;
}

interface IStartPlusServerOptions {
  port?: number;
```

```
development?: boolean;  
}
```

## State

```
// Bookmark state  
AppState.bookmarks.getState(): IBookmark[]  
AppState.bookmarks.setState(bookmarks: IBookmark[]): void  
  
// Widget state  
AppState.widgets.getState(): IWidgetConfig[]  
AppState.widgets.setState(widgets: IWidgetConfig[]): void  
  
// Settings state  
AppState.settings.getState(): ISettings  
AppState.settings.setState(settings: ISettings): void
```

## Interfaces

```
interface IBookmark {  
  id: string;  
  title: string;  
  url: string;  
  description?: string;  
  favicon?: string;  
  folderId?: string;  
  tags?: string[];  
  createdAt: number;  
  updatedAt: number;  
  visitCount: number;  
  lastVisited?: number;  
}  
  
interface IWidgetConfig {  
  id: string;  
  type: string;  
  position: { x: number; y: number };  
}
```

```
size: { width: number; height: number };  
config: Record<string, unknown>;  
enabled: boolean;  
}
```

# License

MIT License - see [license](#) for details.

# Links

- [Repository](#)
- [Issues](#)
- [dees-catalog Documentation](#)

---

Revision #3

Created 2026-03-28 11:15:02 UTC by foss.global Team

Updated 2026-03-28 12:21:47 UTC by foss.global Team