

```

1 # Accurate Player 3 Monorepo
2
3 This monorepo contains all packages related to Accurate Player 3. Packages are located
4 in the `packages` folder:
5
6 - `core` - Core player functionality used by all players
7 - `progressive` - Player that plays videos using [progressive download]
8 (https://en.wikipedia.org/wiki/Progressive\_download)
9 - `hls` - Player that plays videos using [HTTP Live Streaming]
10 (https://en.wikipedia.org/wiki/HTTP\_Live\_Streaming)
11 - `dash` - Player that plays videos using [Dynamic Adaptive Streaming over HTTP]
12 (https://en.wikipedia.org/wiki/Dynamic\_Adaptive\_Streaming\_over\_HTTP)
13 - `abr` - Player that plays both HLS and DASH videos
14 - `cutlist` - Player that plays videos from a cut list using progressive download
15 - `plugins` - Player plugins
16 - `timecode` - Utilities for parsing/manipulating video timecodes
17 - `license-client` - License validation client
18 - `demo` - Player and plugin demos
19
20 `core` contains the basic player logic, interfaces and playback controllers.
21 It relies heavily on logic from `timecode`.
22
23 `core` itself doesn't contain much logic, playback is handled by playback controllers
24 and `plugins` empower the player.
25
26 ## Commands
27
28 - `yarn` - Installs dependencies.
29 - `yarn build` - Builds all packages. Production build. Minified source code and no
30 source maps.
31 - `yarn build:dev` - Builds all packages. Development build. No minification of source
32 code and includes source maps.
33 - `yarn build:watch` - Builds all packages but `demo` in watch mode (rebuilds on file
34 changes). Development build.
35 - `yarn build:docs` - Builds documentation.
36 - `yarn build:demo` - Builds demos.
37 - `yarn test` - Runs all tests
38 - `yarn test:coverage` - Runs all tests and generate [istanbul]
39 (https://istanbul.js.org/) coverage reports.
40 - `yarn lint` - Checks all source for lint errors
41 - `yarn lint:fix` - Checks all source for lint errors and tries to fix all errors that
42 can automatically be fixed.
43 - `yarn start` - Starts builds in watch mode in all packages and starts the `demo`
44 - package on http://localhost:5000
45 - `yarn clean` - Clears all build folders and removes all installed packages.
46
47 See complete list of commands in `package.json`
48
49 **Note** IDE will not find related packages before they've been built.
50
51 # Development
52
53 To start only a subset of the packages with `yarn start` (or any other command) you
54 can leverage `lerna`'s
55 filter options, full documentation [here]
56 (https://github.com/lerna/lerna/tree/main/core/filter-options).
57
58 Some examples:
59
60 Start plugins and the demo package:
61
62 ``
63
64 yarn start --scope "/*/*plugins" --scope "/*/*demo"
65 ``
66
67 Start the progressive player package with all it's dependencies:
68
69 ``
70
71 yarn start --scope "/*/*progressive" --include-dependencies
72 ``
73
74 This feature is useful to not start every package during development to speed things

```

```
up.

60
61 ## JIT Demo
62
63 There are some environment variables used for the Jit demo
64
65 - `JIT_BACKEND`: Specify backend for Jit, can either be a complete url (eg
`http://127.0.0.1:8080` or
66 or just an ip (eg `127.0.0.1`), in which case `http://` and `:8080` is added
automatically.
67 - `AP_KEY`: Specify ap license key to use
68
69 Example command to start: `env JIT_BACKEND=127.0.0.1 AP_KEY=ABCD yarn start`
70
71 ## Test without publishing
72
73 We can test changes locally without the need of publishing changes to nexus npm
repository.
74
75 ### Using npm pack
76
77 Run `npm pack` inside the appropriate package (e.g. packages/core) to create a local
archive `codemill-accurate-player-core-x.y.z.tgz`.
78 Install it where needed by running `npm install path/to/*.tgz`.
79
80 ## Release
81
82 ### Test release
83
84 To create a prerelease unique to your current branch, run:
85
86 ...
87 yarn release:pre
88 ...
89
90 ### Normal release
91
92 Release using
93 `yarn increment -- pre{major|minor|patch}`
94 or just
95 `yarn increment`
96 selecting version manually
97
98 See \[RELEASE.md\] (RELEASE.md)
99
```