**Builder**

**Общий механизм инкрементальной сборки:**

При запуске сборщика всегда ищется файл *lastmtime.json* в котором содержится время модификации последнего файла (*mtime*) и другие файлы-манифесты (*lastmtime.json*, *module-dependencies.json*, *routes-info.json* и др.). При запуске сборщика составляется список файлов, которые будут участвовать в сборке (в таске *01-acc.js* на основании аргумента *modules*). У этих файлов сначала проверяется время последней модификации и если оно меньше чем в *lastmtime.json*, то этот файл не будет участвовать в инкрементальной сборке . Таким образом отсутствие файла *lastmtime.json* будет всегда “провоцировать” не инкрементальную сборку, а полную.

-----------------------------------------------------------------------------------------------------------------------------

**Логика таски *01-acc.js* (acc – сокращенно accumulator):**

Основная задача этого модуля – генерация манифестов (*lastmtime.json*, *module-dependencies.json*, *routes-info.json* и др.) а так же хранение файлов в памяти, которые могут потребоваться для инкрементальной сборки (чтобы не считывать их повторно с диска каждый, например кода делается паковка модулей и один файл может содержаться во многих модулях).

После того как составлен список файлов, которые будут участвовать в сборке они первым делом они проходят через *01-acc.js,* если *mtime* файла больше чем в *lastmtime.json* , тогда в *lastmtime.json* записывается *mtime* файла (после окончания сборки файл *lastmtime.json* сохраняется).

**Логика таски *07-pack.js*:**

Общая логика этой таски такая же как у тасок паковки Grunt-а, но есть один важный момент, который нужно понимать (это касается только паковки для html-файлов). Чтобы осуществить инкрементальную сборку с включенной паковкой нужно хранить базовое состояние html-файла, где прописаны пути до файлов, которые будут паковаться для этой html-ки.

Пример:

Допустим есть файлы *a.js, b.js, c.css, d.css* и *index.html*. В файле *index.html* подключаются все эти файлы по отдельности и выглядит это так:

<script src="a.js"></script>

<script src="b.js"></script>

<link rel="stylesheet" href="c.css">

<link rel="stylesheet" href="d.css">

После паковки (объединения) файлов у нас получится следующее

<script src="a\_b.js"></script>

<link rel="stylesheet" href="c\_d.css">

Теперь мы сохранили эту html-ку. Затем через какое-то время у нас изменился файл *a.js* и если мы не храним исходное состояние html-ки, то мы не узнаем в каком пакете должен быть этот файл. Чтобы знать это, исходное состояние html-файлов сохраняется в файле *packwsmodContents.json*и нам достаточно просто проверить входит ли этот файл в какую-нибудь html-ку и заново сгенерировать пакет для этой html-ки (при этом не будут заново создаваться все пакеты для этой html-ки т.е. в нашем примере js-пакет заново создастся, а css-пакет будет использован тот же).