JAMES internals

Stef van Buuren

2021-05-18

Contents

Preface

This document contains minimal documentation of JAMES internals.

Chapter 1

Introduction

1.1 Overview

This chapter gives a brief overview the Joint Automatic Measurement and Evaluation System (JAMES).

JAMES is an experimental web service for creating and interpreting charts of child growth and development. The current version

1. provides access to high-quality over 300 growth charts used by the Dutch youth health care;
2. interchanges data coded according to the Basisdataset JGZ;
3. screens for abnormal height, weight and head circumference;
4. converts developmental data into the D-score;
5. plot D-scores on special D-score charts;
6. predicts future growth and development.

The service can be used by anyone interested in high-quality charts for monitoring and evaluating childhood growth and development. This chapter highlights the components of JAMES.

1.2 Architecture

JAMES provides its services through OpenCPU, an open system for scientific computing and reproducible research. The system allows for easy integration of growth charts into any `HTTPS` compliant client by means of OpenCPU's API. The JAMES webservice is a RESTful Application Programming Interface (API).

The contents of the system consist of two parts:

- **JAMES**: A collection of R packages that provides back-end functionality
- **JESSE**: A gateway front-end JAMES that translates incoming and outcoming requests

1.3 R packages

1.3.1 JAMES Active packages

Active packages reside on the JAMES server and provide all functionality.

Package	Open	Description
`james`	Y	Joint Automatic Measurement and Evaluation System
`nlreferences`	Y	Growth References for Children living in The Netherlands
`centile`	Y	Translate Measurements, Z-Scores and Centiles with the RIF format
`chartbox`	Y	Collection of Growth Charts
`chartcatalog`	Y	Catalog of JAMES Growth Charts
`chartplotter`	N	Analysing and Plotting Growth Curves
`curvematching`	N	Personalised Prediction by Matching Invididuals
`donorloader`	N	Loads Donor Data from Package or Database
`brokenstick`	Y	Broken Stick Model for Irregular Longitudinal Data
`dscore`	Y	D-Score for Child Development
`bdsreader`	Y	Read Data from the Basisdataset Jeugdgezondheidszorg
`growthscreener`	Y	Finding Children with Unusual Growth Patterns
`jamesclient`	Y	Client-side R Functions for JAMES
`jamesdemodata`	Y	Demo Data for JAMES

1.3.2 JAMES Support packages

Support packages produce half-fabricated materials, provide testing or store documentation.

Package	Open	Description
`donordata`	N	Longitudinal Data for Curve Matching
`chartdesigner`	N	Design Growth Charts for JAMES
`gateway`	Y	Entry to TNO online analytic growth modules
`jamesdocker`	Y	JAMES Docker API
`bdsschema`	Y	Data Exchange Tools for the Basisdataset JGZ
`jamestest`	Y	Shiny app to test JAMES
`minihealth`	Y	Mini Dossier for Individual Health Data
`clopus`	N	Growth reference library

Package	Open	Description
`jamesdocs`	Y	JAMES internals

1.4 JESSE

1.5 JAMES servers

- Production: https://groeidiagrammen.nl/ocpu/test/, Docs (outdated): https://groeidiagrammen.nl
- Test: https://vps.stefvanbuuren.nl/ocpu/test/
- Future: `james.tno.nl`

1.6 Resources

- Demo JAMES at https://tnochildhealthstatistics.shinyapps.io/james_tryout/
- OpenCPU system
- OpenCPU API
- https://www.tno.nl/groei, https://www.tno.nl/growth

Chapter 2

JAMES data format

2.1 Objective

This chapter describes the format of the input data accepted by JAMES. The specification

- closely follows the definition of the Basisdataset JGZ 3.25 (2018);
- defines data objects;
- defines the actions taken by JAMES in case of incorrect, missing or out-of-range data;
- defines the error messages for informing the client.

2.2 Generic object model

2.2.1 EPremDossier Class

2.2.1.1 Object model

EPremDossier	Instance	Class
->	Clientgegevens	EPremGroep
->	Contactmomenten	EPremContactmoment

2.2.1.2 Syntax `C#`

```
public class EPremDossier
```

2.2.1.3 Public properties

Name	Description	Required
Clientgegevens	Class with basic child data	Y
Contactmomenten	Class with data per visit	N
InstrumentCode	Integer identifying the instrument	Ignored
OrganisatieCode	Integer identifying the care organisation	Y
Referentie	String identifying the request	N

2.2.2 EPremGroep Class

2.2.2.1 Object model

EPremGroep	Instance	Class
->	Elementen	EPremElement
->	Groepen	EPremGroep

2.2.2.2 Syntax C#

```
public class EPremGroep
```

2.2.2.3 Public properties

Name	Description	Required
Elementen	Class with BDS-elements	Y
Groepen	Class with groups of BDS-elements	N

2.2.3 EPremElement Class

2.2.3.1 Syntax C#

```
public class EPremElement
```

2.2.3.2 Public properties

Name	Description	Required
Bdsnummer	Integer identifying the BDS-field	Y
InternNummer	Integer identifying internal field	Ignored
Waarde	Value of the BDS-field	Y
Waardeomschrijving	Descriptive label for value	Ignored

2.2.4 EPremContactmoment Class

2.2.4.1 Object model

EPremContactmoment	Instance	Class
->	Elementen	EPremElement
->	Groepen	EPremGroep

2.2.4.2 Syntax `C#`

```
public class EPremContactmoment : EPremGroep
```

2.2.4.3 Public properties

Name	Description	Required
Elementen	Class with BDS-elements	Y
Groepen	Class with groups of BDS-elements	N
Tijdstip	Date of visit	Y

2.3 BDS-elements

BDS	Description	Value	Label	Required
19	Sex of child	“0”	Unknown	Y
		“1”	Male	
		“2”	Female	
		“3”	Not specified	
20	Date of birth	“yyyymmdd”	year-month-day	Y
62	Caretaker relation	“01”	biological father	N
		“02”	biological mother	
		“03”	male partner, stepfather	
		“04”	female partner, stepmother	

BDS	Description	Value	Label	Required
		“05”	adoptive father	
		“06”	adoptive mother	
		“07”	foster father	
		“08”	foster mother	
		“98”	other	
63	Caretaker date of birth	“yyyymmdd”	year-month-day	N
66	Caretaker education	“01”	no primary school	N
		“02”	primary school, special ed	
		“03”	VSO-MLK/IVBO/VMBO-LWOO	
		“04”	LBO/VBO/VMBO-BBL&KBL	
		“05”	MAVO/VMBO-GL&TL	
		“06”	MBO	
		“07”	HAVO/VWO	
		“08”	HBO/HTS/HEAO	
		“09”	WO	
		“98”	Other	
		“00”	Unknown	
71	Caretaker birth country	“dddd”	4-digit code, Table 34	N
82	Gestational age	“ddd”	in days	N
91	Smoking during pregnancy	“1”	yes	N
		“2”	no	
		“99”	unknown	
110	Birth weight	“dddd”	3-4 digits, grammes	N
235	Length/height	“dddd”	3-4 digits, millimeters	N
245	Body weight	“dddddd”	3-6 digits, grammes	N
252	Head circumference	“ddd”	2-3 digits, millimeters	N
238	Height biological mother	“dddd”	3-4 digits, millimeters	N
240	Height biological father	“dddd”	3-4 digits, millimeters	N
510	Passive smoking	“01”	No smoking in house	N
		“02”	Never with child	
		“03”	Not in last 7 days	
		“04”	Yes	

2.4 Error checking

Error checking of the JSON data occurs in three phases:

1. PHASE 1: Check whether the JSON data are valid JSON. The process terminates with an error message if the input JSON is not valid.

2. PHASE 2: Validate the JSON data against the JSON schema specification. The process terminates with an error if any required fields are miss-

ing. The process generates messages for data points that do not conform to the JSON schema, but continues.

3. PHASE 3: Check the range of the numeric data. The process generates messages for out-of-range values, but continues using the specified values.

The default JSON schema in phase 2 is the built-in JSON schema `bds_schema_str.json`, a data format implementing a version that accepts strings as values for BDS-elements.

Chapter 3

Growth charts in JAMES

3.1 Chart naming conventions

The link https://groeidiagrammen.nl/ocpu/lib/james/www/ contains an interactive overview of the available growth charts. There are 342 different charts: for boys and girls, for preterms, for different age ranges, for specific ethnic groups, for height, weight, BMI, and so on. Each chart has a chart code, a character code identifying the design. This section explains the construction of the chart codes.

The GitHub repository https://github.com/stefvanbuuren/chartbox contains the chart libraries that are available to JAMES. The `list_charts()` function produces a tabular overview.

```
charts <- chartbox::list_charts()
dim(charts)
```

```
## [1] 394   8
```

```
charts[c(1, 22, 23, 300, 301, 340), ]
```

```
##     chartgrp chartcode population    sex design  side language week
## 1     nl2010      HJAA         HS   male      A front    dutch     
## 22    nl2010      HMBH         HS female      B   hgt    dutch     
## 23    nl2010      HMBR         HS female      B   wfh    dutch     
## 300  preterm   PMAHN28         PT female      A   hgt    dutch   28
## 301  preterm   PMAHN29         PT female      A   hgt    dutch   29
## 340  preterm   PMEAN32         PT female      E front    dutch   32
```

The `chartbox` package currently contains three chart groups: `nl2010`, `preterm` and `who`. Each group collects charts of a similar type.

Chart Group	Charts	Chart code	Description	Source
nl2010	136	CCCC	Dutch children 0-21 years, including minorities	Talma et al. (2010)
preterm	192	CCCCCNN	Dutch preterms, ga <= 36 weeks, 0-4 years	Bocca-Tjeertes et al. (2012
who	14	CCCC	WHO Child Growth Standards 0-4 years	WHO

The chart code is an alpha-numeric code of four (for nl2010 and who) or seven (for preterm) that uniquely identifies each of the charts. The table below specifies the full coding schema used to construct the chart codes.

Position	Field	Value	Description
1	Population	N	Dutch
		T	Turkish
		M	Moroccan
		H	Hindostan
		P	Preterm
		W	WHO
2	Sex	J	Male
		M	Female
3	Design	A	0-15 months
		B	0-4 years, WFH
		C	1-21 years
		D	0-21 years
		E	0-4 years, WFA
4	Side	A	A4, front
		B	A4, back
		C	A4, back, no hdc
		H	square, hgt
		O	square, hdc
		Q	square, bmi
		R	square, wfh
		W	square, wgt
		X	A4, double sided
5	Language	N	Dutch
		E	English
6-7	Week	25-36	Gestational age

For illustration, code NJAA references to Dutch (N), boys (J), 0-15 month (A), front side (A). Likewise, PMEAN33 codes for the chart of preterm (M), girls (M), 0-4 years (E), front side (A), Dutch language (N) born at 33 weeks of gestation (33).

Some forms hold multiple growth charts. For example, the NJAA chart is designed for A4 paper size (297mm × 210mm) and contains three growth charts:

head circumference by age, length by age, and weight by age. Some others have no diagram, like NJAB with explanations. All square formats hold one growth chart. All of the square forms have equal sizes (160mm × 160mm).

The following table lists the measures per design-form combination.

Design	Side	Measure	Description
A	A	hdc	Head circumference by age, 0-15 mo
		hgt	Length by age, 0-15 mo
		wgt	Weight by age, 0-15 mo
	B		
	H	hgt	Length by age, 0-15 mo
	O	hdc	Head circumference by age, 0-15 mo
	W	wgt	Weight by age, 0-15 mo
B	A	wfh	Weight for height, 0-4 yr
		hgt	Length by age, 0-4 yr
	B	hdc	Head circumference by age, 0-4 yr
	C		
	H	hgt	Height by age, 0-4 yr
	O	hdc	Head circumference by age, 0-4 yr
	R	wfh	Weight for height, 0-4 yr
	W	wgt	Weight by age, 0-4 yr
C	A	wfh	Weight for height, 1-21 yr
		hgt	height by age, 1-21 yr
	B	bmi	BMI by age, 1-21 yr
		hdc	Head circumference by age, 1-21 yr
	C	bmi	BMI by age, 1-21 yr
	H	hgt	Height by age, 1-21 yr
	O	hdc	Head circumference by age, 1-21 yr
	Q	bmi	Body mass index by age, 1-21 yr
	R	wfh	Weight for height, 1-21 yr
E	A	wgt	Weight by age, 0-4 yr
		hgt	height by age, 0-4 yr
	B	hdc	Head circumference by age, 0-4 yr
	H	hgt	Height by age, 0-4 yr
	O	hdc	Head circumference by age, 0-4 yr
	W	wgt	Weight by age, 0-4 yr

Chapter 4

Methods

We describe our methods in this chapter.

Chapter 5

D-score implementation

This document describes the actions needed to implement the D-score into JAMES. The functionality of JAMES is distributed over multiple packages. This set of actions may be of interest when implementing new features.

5.1 Actions

Package	PR	Description
minihealth	03a32f1	Create milestones descriptions
dscore	f0013ce	Link BDS number to Van Wiechen milestones
dscore	6886854	Fine tuning of milestone labels
minihealth		Create the `bds_lexicon` object
minihealth	4893982	Add milestones to BDS validation JSON schema
minihealth	0069671	Add `convert_ddi_gsed()` to convert BDS-milestones into GSED items
minihealth	8ab1392	Add a new class `individualDS` for storing milestones, D-score and DAZ
clopus	1182cb0	Add Dutch and GCDG D-score references

Package	PR	Description
clopus	7bdbcd9	Construct age-shifted D-score references for preterms
clopus	ceab7f9	Import the D-score references into `clopus`
chartdesigner	6883190	Add chart constructor functions for D-score, both terms and pre-terms
chartdesigner	511f456	Extend internal `set.axes.design()` to D-score charts
chartdesigner	6582af8	Extend to `axes.locations` object to D-score charts
chartdesigner	47e3cc3	Create `dchart()` function and extend its helper functions
chartdesigner	fbbc7c8	Function `chartcode()` factory, make one function for each chart code
chartcatalog	cc46788	Extend the chart naming system to D-score charts
chartcatalog	84aaded	Extend the lookup table `ynames_lookup` to handle new D-score charts
chartbox	aa31067	Extend chart box with all D-score charts
james	6412840	Add radio button for D-score charts
minihealth	06a04c9	Calculate D-score and DAZ
chartplotter	4b58638	Skip the `dsc` field for finding matches
minihealth	816be33	Add D-score and DAZ to class `individualAN`
donordata	77e01b4	Add milestones to SMOCC donor data
donordata	ecb3413	Calculate D-score and DAZ for SMOCC data
donordata	3fa9d4d	Fit and store brokenstick model for D-score on SMOCC data
donorloader	c22c446	Update internal data after changes in donordata
jamesdocs	TBD	Document steps (this file)

Package	PR	Description
donordata	7983c3	Saves the item scores to create JSON files
donordata	1537182	Save mapping between SMOCC and BDS coding scheme
donorloader	e9a8ed	Make `smocc_bds` available to JAMES
jamestest	648419	Regenerate smocc JSON files to include DDI scores
jamestest	ce1dbe	Update the `installed.cabinets` object with the new individual milestones data
minihealth	4dda8d	Add class `individualRW` to store and convert raw milestones data
minihealth	9e03e7	Complete the JSON validator schema

Chapter 6

Dockerfile for JAMES

6.1 Objective

This chapter describes how to build and deploy JAMES as a Docker container.

6.2 Pre-requisites

JAMES is currently constructed from a collection of `R` packages. The top-level package at https://github.com/stefvanbuuren/james also defines a Javascript interface in the `inst/www` directory. Deployment of JAMES relies on the `OpenCPU` server. In principle, it is enough to install the `james` package on the `OpenCPU` server, and will also install all dependencies.

The following is needed to build and run a JAMES image:

- Permission to read from the following private repo's:
  - `stefvanbuuren/chartplotter`
  - `stefvanbuuren/clopus`
  - `stefvanbuuren/curvematching`
  - `stefvanbuuren/donorloader`
- Personal Github token with repo scope from here, Generate a token with only scope repo.
- Install `Docker Desktop` on your local machine, and run some tutorials

6.3 Dockerfile

```
FROM opencpu/base as intermediate

# install for V8 package
RUN apt-get update && apt-get install -y libnode-dev

# repo_token.txt should be something like "GITHUB_PAT=624adeaa..."
# to be able to install packages from private repo's
COPY repo_token.txt .Renviron

# install R packages needed for JAMES
RUN \
    R -e 'install.packages("remotes")' \
    R -e 'remotes::install_github("stefvanbuuren/clopus")' \
    R -e 'remotes::install_github("stefvanbuuren/dscore")' \
    R -e 'remotes::install_github("stefvanbuuren/chartbox")' \
    R -e 'remotes::install_github("stefvanbuuren/brokenstick")' \
    R -e 'remotes::install_github("stefvanbuuren/minihealth")' \
    R -e 'remotes::install_github("stefvanbuuren/jamestest")' \
    R -e 'remotes::install_github("stefvanbuuren/growthscreener")' \
    R -e 'remotes::install_github("stefvanbuuren/james")'

# rebuild layer without .Renviron
FROM opencpu/base

LABEL maintainer="stef.vanbuuren@tno.nl"

# re-install for V8 package
RUN apt-get update && apt-get install -y libnode-dev

# copy R libraries
COPY --from=intermediate /usr/local/lib/R/site-library /usr/local/lib/R/site-library

# modify preload section
COPY james.conf /etc/opencpu/server.conf.d/james.conf

# install Arial
RUN mkdir /usr/share/fonts/truetype/arial/
COPY fonts/truetype/arial/* /usr/share/fonts/truetype/arial/

CMD service cron start && apachectl -DFOREGROUND
```

- Create a fresh directory, and `cd` to this directory in your terminal

- Store the script given above in a file named `Dockerfile`
- Create a file named `repo_token.txt` with one line: `GITHUB_PAT=624adeaa...` with your token
- Create `fonts/truetype/arial` with the proper `ttf` files
- Create a file `james.conf` specifying the preload libraries

```
{
    "enable.api.library": true,
    "enable.api.apps": true,
    "enable.api.user": true,
    "enable.api.tmp": true,
    "enable.cors" : true,
    "enable.post.code": true,
    "enable.rlimits" : true,
    "error.showcall": true,
    "smtp.server" : "localhost",
    "smtp.use.ssl" : "no",
    "httpcache.post": 300,
    "httpcache.lib": 86400,
    "httpcache.apps": 900,
    "httpcache.tmp": 86400,
    "httpcache.static": 31536000,
    "key.length" : 13,
    "repos": "https://cran.rstudio.com",
    "rlimit.as": 4e9,
    "rlimit.fsize": 1e9,
    "rlimit.nproc": 500,
    "timelimit.get": 60,
    "timelimit.post": 90,
    "timelimit.webhook": 900,
    "preload": ["clopus", "donorloader", "chartbox", "brokenstick", "minihealth", "growthscreener
}
```

6.4 Docker commands

Build the `james` image, type in a terminal

```
docker build -t james .
```

This may takes a long time (30 minutes), in which the entire application is downloaded from various web-locations. After (hopefully succesful) completion, check the image

```
docker images -a
```

If all is well, the top line is called `james`. Now run the container on your local machine:

```
docker run -t -d -p 80:80 james
```

If the ports are already taken by other containers, stop and remove all containers:

```
docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)
```

Reissue the `docker run`, and the container should now run. Check by

```
docker ps
```

which should list a container created from the `james` image.

If you want to enter the container use

```
docker exec -i -t 6c /bin/bash
```

where `6c` are the first two characters of the container ID.

Inside the container, check font matching of Arial as

```
fc-match Arial
```

6.5 Checks with the browser

```
http://localhost
```

should show Apache2 Ubuntu default screen.

```
http://localhost/ocpu/test/
```

should show `OpenCPU` test page.

```
http://localhost/rstudio/
```

should start the Rstudio IDE - *if installed* . Use `opencpu:opencpu` to log in.

```
http://localhost/ocpu/library/james/www/
```

should start the JAMES javascript interface.

See also https://registry.hub.docker.com/r/opencpu/rstudio

6.6 Security

1. Don't use the intermediate container, since it will contain your token in `/.Renviron`. The latest (`james`) container does not hold your token, and can be shared.
2. The container is shielded from the machine on which it runs. However, the materials within the container are only protected by `R_LIMITS`. In general, for production it is wise to add restriction on the `OpenCPU` server.

Bibliography

Bocca-Tjeertes, I., van Buuren, S., Bos, A., Kerstens, J., ten Vergert, E., and Reijneveld.S.A. (2012). Growth of preterm and fullterm children aged 0-4 years: Integrating median growth and variability in growth charts. *Journal of Pediatrics*, 161(3):460–465.

Talma, H., Schonbeck, Y., Bakker, B., Hirasing, R., and van Buuren, S. (2010). *Groeidiagrammen 2010: Handleiding bij het meten en wegen van kinderen en het invullen van groeidiagrammen.* TNO Kwaliteit van Leven, Leiden.