

10° ROW – Online, June 8<sup>th</sup>, 2021



# jscontact-tools

Mario Loffredo  
IIT-CNR/Registro.it

# JSContact

- A specification defining a data model and JSON representation of contact card information
  - Alternative to vCard data format
  - Unambiguous, extensible and simple to process
  - Defines Card and CardGroup objects
- Related documents:
  - ***R. Stepanek, M. Loffredo: «JSContact: a JSON representation of contact data», [draft-ietf-jmap-jscontact](#)***
  - ***M. Loffredo, R. Stepanek: «JSContact: Converting from and to vCard», [draft-ietf-jmap-jscontact-vcard](#)***
  - ***M. Loffredo, G. Brown: «Using JSContact in Registration Data Access Protocol (RDAP) JSON Responses», [draft-ietf-regext-rdap-jscontact](#)***

# JSContact – common data types

- **ExtensibleEnum**: a string whose value is pre-defined, registered in a future RFC or vendor-specific (e.g. Context, Kind, StreetComponent, PhoneFeature)
- **Id**: a string whose allowed characters are: ("A-Za-z0-9"), hyphen ("-"), and underscore ("\_"). It is used to represent a map key
- **UTCDateTime**: a string based on RFC 3339 "date-time" format with some further restrictions
- **LocalizedString**: a data structure allowing for internationalisation of string values
- Collections:
  - **A[B]**: A JSON object where the keys are all of type "A", and the values are all of type "B";
  - **A[ ]**: An array of values of type "A".

# JSContact – other data types

- **NameComponent:** a component of the entity name;
- **Address:** a physical location of the entity;
- **StreetComponent:** a street address component (e.g. name, number, building, floor)
- **Phone:** a phone number to contact the entity;
- **EmailAddress:** an email address to contact the entity;
- **Resource:** an online resource (e.g. a web url, a social media account) of the entity;
- **File:** a photograph or an image of the entity;
- **ContactLanguage:** a language for contacting the entity;
- **Organization:** companies or organization names and units of the entity;
- **Title:** a job title or functional position of the entity;
- **Relation:** the relationships between the entity and other entities;
- **Anniversary:** a memorable date or event for the entity;
- **PersonalInformation:** personal information about the entity (e.g. hobby).

# JSContact – Card properties

- String uid;
- String prodId;
- UTCDateTime created;
- UTCDateTime updated;
- ExtensibleEnum kind;
- String[Relation] relatedTo;
- NameComponent[ ] name;
- LocalizedString fullName;
- LocalizedString[ ] nickNames;
- Id[Organization] organizations;
- Id[Title] titles;
- Id[EmailAddress] emails;
- Id[Phone] phones;
- Id[Resource] online;
- Id[File] photos;
- Enum preferredContactMethod;
- String[ContactLanguage[ ]] preferredContactLanguages;
- Id[Address] addresses;
- Id[Anniversary] anniversaries;
- Id[PersonalInformation] personalInfo;
- LocalizedString notes;
- String[Boolean] categories;

# JSContact – CardGroup properties

- String uid;
- String[Boolean] members;
- String name;
- Card card;

# jscontact-tools (1)

- An open-source Java library for JSContact:
  - creation;
  - validation;
  - serialization/deserialization;
  - conversion from and to vCard (RFC 6350), xCard (RFC 6351) and jCard (RFC 7095)
- other RFCs involved: 6473, 6474, 6715, 6879, 8605

# jscontact-tools (2)

- Makes use of annotations (lombok, com.fasterxml.jackson, javax.validation)
- Leverages the features of the ez-vcard project (<https://github.com/mangstadt/ez-vcard>)
- Performs ~ 340 tests
- Last version: 0.4.1
- Compliant with jscontact-06 and jscontact-vcard-04
- **Source:** <https://github.com/consigliNazionaleDelleRicerche/jscontact-tools>
- **Jar:** <https://mavenrepository.com/artifact/it.cnr.iit.jscontact/jscontact-tools>



# jscontact-tools – creation

- Achieved through Lombok builders
- Allows simplest maps loading by putting entries singularly

```
EmailAddress email = EmailAddress.builder()
    .context(Context.work(), Boolean.TRUE)
    .context(Context.private(), Boolean.TRUE)
    .email("mario.loffredo@iit.cnr.it")
    .build();
```

- Checks for presence of required properties

```
//email missing - NullPointerException is raised
EmailAddress.builder().context(Context.work(), Boolean.TRUE).build();
```

- Also implemented through the cloning of top most objects

# jscontact-tools – validation

- Applied to top most objects
- The method `isValid` returns `true`, if all the constraints are satisfied, `false`, otherwise
- The method `getValidationMessages` returns the list of error messages:

```
@Test public void testInvalidCountryCode() {  
  
    // a country code must be two-character long  
    Map addresses = new HashMap<String,Address>(){{ put("ADR-1",  
    Address.builder().countryCode("ita").build()); }};  
    Card jsCard = JSCard.builder().uid(getUUID()).addresses(addresses).build();  
  
    assertTrue("testInvalidCountryCode-1", !jsCard.isValid());  
    assertTrue("testInvalidCountryCode-2", jsCard.getValidationMessage().equals("invalid countryCode in  
    Address"));  
  
}
```

# jscontact-tools – vCard validation

- JCardValidator
  - `void validate(String json)`
  - `void validate(com.fasterxml.jackson.databind.JsonNode jsonNode)`
- VCardValidator
  - `void validate(String vcf)`
- XCardValidator
  - `void validate(String xml)`

# jscontact-tools – serialization/deserialization

- Performed through Jackson library annotations
- Pretty printing of serialized JSContact objects is supported
- Some custom serializers/deserializers have been implemented to handle:
  - ExtensibleEnum values;
  - UTCDateTime values;
  - Deserialization of a cards collection into a list of polymorphic objects (i.e. Card and CardGroup instances)

# jscontact-tools – vCard <-> JSContact conversion

- EZVCard2JSContact  
`List<JSContact> convert(List<VCard>)`
- VCard2JSContact  
`List<JSContact> convert(String vcf)`
- JCard2JSContact  
`List<JSContact> convert(String json)`  
`List<JSContact> convert(JsonNode jsonNode)`
- XCard2JSContact  
`List<JSContact> convert(String xml)`
- JSContact2EZVCard  
`List<VCard> convert(List<JSContact>)`  
`List<VCard> convert(String json)`
- JSContact2VCard  
`String convertToText(List<JSContact>)`
- JSContact2JCard  
`String convertToJson(List<JSContact>)`  
`JsonNode convertToJsonNode(List<JSContact>)`
- JSContact2XCard  
`String convertToXml(List<JSContact>)`

# jscontact-tools – conversion example

```
[
  ["version", {}, "text", "4.0"],
  ["fn", {}, "text", "Joe User"],
  ["kind", {}, "text", "individual"],
  ["org", {"type": "work"}, "text", "Example"],
  ["adr", {"type": "work"}, "text",
    [
      "", "Suite 1234", "4321 Rue Somewhere", "Quebec",
      "QC", "G1V 2M2", "Canada"
    ]
  ],
  ["tel", {"type": ["work", "voice"]}, "uri", "tel:+1-555-555-1234"],
  ["email", {"type": "work"}, "text", "joe.user@example.com"]
]
```

```
{
  "uid": "4f0a5931-9cca-4c5d-b9bc-be73317ae2e8",
  "fullName": {"value": "Joe User"},
  "kind": "individual",
  "organizations": { "ORG-1": {"name": {"value": "Example"}} },
  "addresses": {
    "ADR-1" : {
      "contexts": {"work": true},
      "street": [ {"type": "name", "value": "4321 Rue Somewhere"},
                  {"type": "extension", "value": "Suite 1234"} ],
      "locality": "Quebec",
      "region": "QC",
      "postcode": "G1V 2M2",
      "country": "Canada"
    }
  },
  "phones": { "PHONE-1": {
    "contexts": {"work": true},
    "features": {"voice": true},
    "phone": "tel:+1-555-555-1234"
  }
  },
  "emails": { "EMAIL-1": {
    "contexts": {"work": true},
    "email": "joe.user@example.com"
  }
  }
}
```

# jscontact-tools – dealing with extensions

- Extensible types

```
EmailAddress email = EmailAddress.builder()
    .context(Context.work(), Boolean.TRUE)
    .context(Context.ext("institutional"), Boolean.TRUE)
    .email("mario.loffredo@cnr.it")
    .build();
```

- Customizations

- Must be prefixed with a domain name controlled by the vendor to avoid conflicts:

- properties (e.g. "example.com/customprop");

- values (e.g. "example.com/customvalue").

- Within vCard-JSContact conversion, the name of unmatched vCard components is prefixed with “ietf.org/ <RFC defining the component>/”

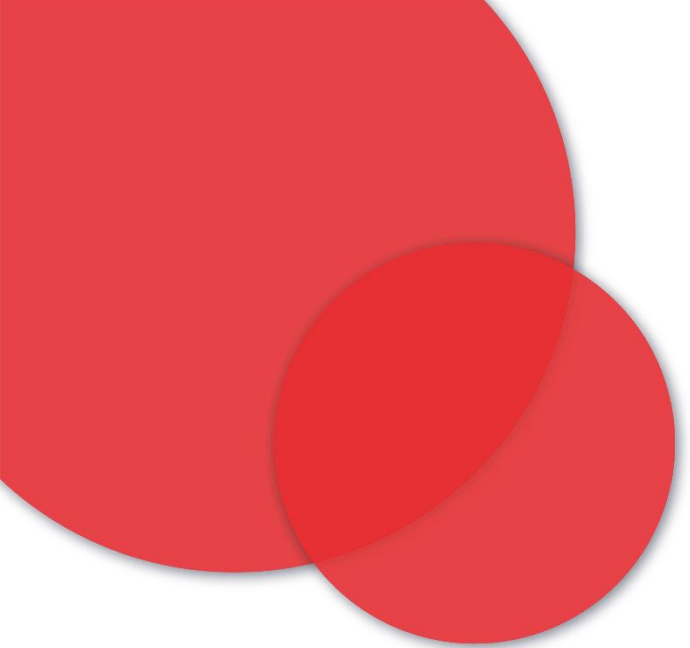
# Possible developments

- Reflect changes to draft-ietf-jmap-jscontact(-vcard)
- Specification is quite stable
- Last open issue is about how to handle language-dependent information:
  - **Option 1.** Keep supporting it through the LocalizedString type
  - **Option 2.** Use a JSONPatch-like approach



# Feedback

- HOW TO:
  - Raising an issue on <https://git.tools.nic.it/rdap/jscontact-tools/-/issues>
  - Sending an email to <mailto:mario.loffredo@iit.cnr.it>



**Thanks for your attention!**  
Q&A