C2E Specification

1. Introduction

C2E defines a distribution and interchange format for digital publications and documents. The C2E format provides a means of representing, packaging and encoding structured and semantically enhanced Web content — including HTML, CSS, SVG and other resources — for distribution in a single-file container.

The expectation is that the C2E format will be utilized for a broad range of content, including educational, professional and scientific publications.

This initial release aims to support a wide range of interactive digital requirements, including complex layouts, rich media and interactivity, and global typography features.

C2E is modular in nature: it consists of a family of specifications that define the core features and functionality of the standard. This specification represents the primary entry point to standard, but the specifications listed in Specifications are all a part of C2E. An index to key concepts and definitions defined across these specifications is provided at the end of this specification.

2. C2E Conceptual Model

2.1 Major Elements

The C2E Package depicted in Figure 2.1 consists of two major elements: a special JSON-LD file describing the content organization and resources in a Package, and the file resources being described by the JSON-LD. The special JSON-LD file is called the C2E Manifest file, because course content and organization is described in the context of ‘manifests’. Once a Package has been incorporated into a single file for transportation, it is called a Package Interchange File. The relationship of these parts to the content container is described below:

Package

It is a logical directory, which includes a specially named JSON-LD file and contains the actual file resources. The file resources may be organized in sub-directories.

Package Interchange File

It is a single file, (e.g., ‘.zip’, ‘.jar’, ‘.c2e’) which includes a top-level manifest file named “c2e.json” and all other files and directories as identified by the Manifest. A Package Interchange File is a concise Web delivery format, a means of transporting related, structured information. PKZip v2.04g (.zip) is recommended as the default Package Interchange File format. Any ZIP file format MUST conform to RFC1951.

2.2 Standard Name for the Manifest File

Content distributed according to the C2E specification must contain an C2E Manifest file. To ensure that the C2E Manifest file is a valid JSON-LD file and can always be found within a Package, it has a pre-defined name and location:

c2e.json

In the absence of this file, the package is not an C2E Package and cannot be processed. It is required that the name be kept, as above, in all lowercase letters.

3. JSON-LD

JSON-LD is a lightweight Linked Data format. It is easy for humans to read and write. It is based on the already successful JSON format and provides a way to help JSON data interoperate at Web-scale. JSON-LD is an ideal data format for programming environments, REST Web services, and unstructured databases.

Documentation - JSON-LD 1.1

3.1 Relationship to RDF (Resource Description Framework)

JSON-LD is a concrete RDF syntax. Hence, a JSON-LD document is both an RDF document and a JSON document and correspondingly represents an instance of an RDF data model which is set of triples, each consisting of a subject, a predicate and an object.

4. Manifest Elements

A manifest is a description in JSON-LD of the resources comprising meaningful instruction for a C2E. A manifest also contain zero or more static ways of organizing the instructional resources for presentation.

The scope of manifest is elastic. A manifest can describe part of a C2E that can stand by itself outside of the context of a C2E (an instructional object). The decision is given to content developers to describe their content in the way they want it to be considered for aggregation or disaggregation. The general rule is that a Package always contains a single top-level manifest that may contain one or more sub-Manifests. The top-level manifest always describes the Package. Any nested sub-Manifests describe the content at the level to which the sub-Manifest is scoped, such as a course, instructional object, or other.

4.1 The Context

One of basic concept that JSON-LD has is the Context allow us to define terms that can be used to define the components and resources of the C2E. A context is introduced using an entry with the key `@context` and may appear within a node object or a value object.

```
{
  "@context": {

  }
}
```

4.1.1 Namespaces

Define following namespace prefixes are defined as context terms:

`"c2ens": "https://c2e.curriki.org/"`

`"sdons": "https://schema.org/"`

`"xsd": "http://www.w3.org/2001/XMLSchema/#"`

`c2ens` namespace prefix is defined to refer the resources, types and terms by C2E.

`sdons` namespace prefix is defined to refer vocabularies by Schema.

`xsd` namespace prefix define XML Schema language which also referred as XML Schema Definition (XSD).

The context with required namespaces would look like:

```
{
  "@context": {
    "c2ens": "https://c2e.curriki.org/",
    "sdons": "https://schema.org/",
    "xsd": "http://www.w3.org/2001/XMLSchema/#"
  }
}
```

4.1.2 C2E Mandatory Terms

Following mandatory terms are required to define in the JSON-LD `@context`.

```
{
  "@context": {
    "c2ens": "https://c2e.curriki.org/",
    "sdons": "https://schema.org/",
    "xsd": "http://www.w3.org/2001/XMLSchema/#",
    "C2E": "sdons:CreativeWork",
    "c2eContainer": {"@id": "sdons:Collection", "@container": ["@graph", "@index"]},
    "c2eResources": "@graph",
    "c2eContentTypes": "@graph",
    "c2eContents": "@graph",
    "c2eEmbeds": "@graph",
    "c2eTerm": "c2ens:terms/",
    "c2eVendor": {"@id": "sdons:vendor", "@type":"xsd:string"},
    "name": {"@id": "sdons:name", "@type":"xsd:string"},
    "url": {"@id": "sdons:url", "@type":"xsd:string"},
    "fileFormat": {"@id": "sdons:fileFormat", "@type":"xsd:string"},
    "@language" : "en"
  }
}
```

`"C2E": "sdons:CreativeWork"` The `C2E` term is used to define primary attribute `@type` of JSON-LD top level object.

`"c2eEmbeds": "@graph"` The `c2eEmbeds` term is use to represent the set of associated components (that can also be a C2E Learning Experiences) of a C2E. It is basically a JSON-LD's `@graph`.

`"c2eResources": "@graph"` is an important attribute that is used to define the collection if resource files for example an html, text, image, audio, video or scripts. Each resource is consist of two main attributes `url` and `fileFormat`.

`"c2eContentTypes": "@graph"` Is used to define *Content Types* which is explained further in the document.

`c2eContainer` term is used to define a container object that holds to primary structures `c2eResources` and `c2eEmbeds` mentioned above.

`"name": {"@id": "sdons:name", "@type":"xsd:string"}` The `name` term is used to define the name of an object.

`c2eTerm` is `c2ens` namespace extension to be used as prefix of C2E specific IRIs in the JSON-LD structure.

`c2eVendor` is use to define the technology vendor who enable the platform to make C2E based on their technology. For example H5P etc.

`@language` keyword in `@context` define the default language of the manifest JSON-LD document.

4.1.3 C2E Metadata Terms

Define following terms in the `@context` which is used to form a `c2eMetadata` object:

```
"c2eMetadata": "c2eTerm:c2eMetadata",
"schemaVersion": {"@id": "sdons:schemaVersion", "@type":"xsd:string"},
"general": "c2eTerm:mdGeneral",
"subjectOf": "sdons:subjectOf",
"title": {"@id": "sdons:headline", "@type":"xsd:string"},
"description": {"@id": "sdons:description", "@type":"xsd:string"},
"keywords": {"@id": "sdons:keywords", "@type":"xsd:string"},
"author": {"@id": "sdons:author", "@type":"xsd:string"},
"publisher": {"@id": "sdons:publisher", "@type":"xsd:string"},
"email": {"@id": "sdons:email", "@type":"xsd:string"},
"lifecycle": "c2eTerm:mdLifecycle",
"version": {"@id": "sdons:version", "@type": "xsd:string"},
"releaseStatus": {"@id": "sdons:creativeWorkStatus", "@type": "xsd:string"},
"subManifest": {"@id": "sdons:url", "@type":"xsd:string"},
"copyright": "c2eTerm:copyright",
"license": {"@id": "sdons:license", "@type": "xsd:string"},
"encodingFormat": {"@id": "sdons:encodingFormat", "@type": "xsd:string"},
"copyrightHolder": {"@id": "sdons:copyrightHolder", "@type": "xsd:string"},
"copyrightNotice": {"@id": "sdons:copyrightNotice", "@type": "xsd:string"},
"copyrightYear": {"@id": "sdons:copyrightYear", "@type": "xsd:integer"}
```

4.2 Define C2E JSON-LD Object

4.2.1 Define Metadata

Make a following `c2eMetadata` object right after defining all terms in `@context` structure:

```
"c2eMetadata": {
    "@id": "c2ens:c2eid-xxx/c2eMetadata",
    "@type": "sdons:Dataset",
    "schemaVersion": "1.0",
    "general": {
      "@id": "c2ens:c2eid-xxx/c2eMetadata/general",
      "@type": "sdons:Dataset",
      "title": "My Smaple C2E",
      "description": "My Smaple C2E for Proof of Concept.",
      "keywords": "c2e, e-learning, learning experince"
    },
    "subjectOf": {
      "@id": "c2ens:c2eid-xxxxxx/metadata/subjectOf",
      "@type": "sdons:CreativeWork",
      "name": "Anatomy and Physiology All-in-One For Dummies®"
    },
    "author" : {
      "@id": "c2ens:c2eid-xxx/author/id/xxx",
      "@type": "sdons:Person",
      "name": "Waqar Muneer",
      "email": "waqar@curriki.org",
      "url": "https://twitter.com/waqarmnr"
    },
    "publisher" : {
      "@id": "c2ens:c2eid-xxx/author/id/xxx",
      "@type": "sdons:Organization",
      "name": "Curriki",
      "email": "info@curriki.org",
      "url": "https://curriki.org"
    },
    "lifecycle": {
      "@id": "c2ens:c2eid-xxx/c2eMetadata/lifecycle",
      "@type": "sdons:Code",
      "version": "1.10",
      "releaseStatus": "Beta"
    },
    "copyright": {
      "@id": "c2ens:c2eid-xxx/c2eMetadata/copyRight",
      "@type": "sdons:Dataset",
      "license": {
        "@id": "c2ens:c2eid-xxx/license/id/xxx",
        "@type": "sdons:DigitalDocument",
        "encodingFormat": "text/plain or any binary as per need",
        "url": "/path/to/license.txt",
        "fileFormate": "text/plain",
        "licensee": {
            "@id": "c2ens:c2eid-7ffce150-404e-11ee-9cff-c161055bb056/licensee/id/waqar@curriki.org",
            "@type": "sdons:Person",
            "name": "Waqar",
            "email": "waqar@curriki.org",
            "url": "https://x.com/waqarmnr"
        }
      },
      "copyrightHolder": {
        "@id": "c2ens:c2eid-xxx/copyrightHolder/id/xxx",
        "@type": "sdons:Person",
        "name": "Waqar Muneer",
        "email": "waqar@curriki.org",
        "url": "https://twitter.com/waqarmnr"
      },
      "copyrightNotice": "This C2E has all rights to Waqar Muneer",
      "copyrightYear": "2023"
    }
}
```

4.2.2 Define C2E

C2E consist of `"@id"`, `"@type"`, `"name"` mandatory attributes.

It makes JSON-LD top level object. For example:

```
{
    "@context": {
      "c2ens": "https://c2e.curriki.org/",
      "sdons": "https://schema.org/",
      "xsd": "http://www.w3.org/2001/XMLSchema/#",
      "C2E": "sdons:CreativeWork"
      ........................
      ........................
    },
    "c2eMetadata": {
      "@id": "c2ens:c2eid-xxx/c2eMetadata",
      "@type": "sdons:Dataset",
      "schemaVersion": "1.0"
      ........................
      ........................
    },
    "@id": "c2ens:c2eid-xxx",
    "@type": "C2E"
  }
```

4.2.2.1 C2E Identifier

C2E identifier which is defined as JONS-LD `@id` is required to have following pattern:

`c2ens` `:` `c2eid-` `xxx`

which will be translated as:

https://c2e.curriki.org/c2eid-xxx

`xxx` represent the system generated unique number.

4.2.2.2 C2E Status

Since C2E object is `@type` of `CreativeWork`, so its status can is defined using `creativeWorkStatus` property by following schema.org standards.

`creativeWorkStatus` has following possible values `preview` and `published`.

```
{
  "@id": "c2ens:c2eid-xxx",
  "@type": "C2E",
  "creativeWorkStatus": "preview"
}
```

4.2.3 Define C2E Resources

Includes references to all of the resources included in the package. At a minimum it should reference all those files that are needed in order to view the content.

In C2E JSON-LD `c2eContainer` container object, an object is defined with `@id` and `c2eResources` attribute which is used to list all resources that are required for a C2E.

```
"c2eContainer": [
    {
      "@id": "c2ens:c2eResources",
      "@type": "sdons:Collection",
      "c2eResources" : [
        {
          "@id": "c2ens:c2eid-xxx/resource/1",
          "@type": "sdons:DigitalDocument",
          "url": "/path/to/file-1.json",
          "fileFormat": "application/json"
        },
        {
          "@id": "c2ens:c2eid-xxx/resource/2",
          "@type": "sdons:DigitalDocument",
          "url": "/path/to/readme.txt",
          "fileFormat": "application/text"
        },
        {
          "@id": "c2ens:c2eid-xxx/resource/book.epub",
          "@type": "sdons:DigitalDocument",
          "url": "/resources/book.epub",
          "fileFormat": "epub",
          "identifier": {
              "@type": "sdons:PropertyValue",
              "propertyID": "ISBN",
              "value": "123456789"
          }
        }
      ]
    }
  ]
```

4.4 Extending C2E Object with “hasPart” attribute (optional)

Since C2E Object is *sdons* based “CreativeWork, so C2E object is considered to be have all of its attributes predefined in it. A C2E Object can be extended by using *“hastPart”* attribute to associate other “CreativeWork” type of object. CreativeWork’s *sub-types* such as Article, Blog , LearningResource etc can also be assigned to *“hastPart”* attribute.

```
{
  "@id": "c2ens:c2eid-xxx",
  "@type": "C2E",
  "hasPart": {
    "@id": "c2ens:c2ePart-xxx",
    "@type": [ "Any specific type of sdons:CreativeWork" ]
  }
}
```

“hastPart” attribute provide ability to create CreativeWork and its *sub-types* based nested structure.

5. Defining C2E Content Types

Since C2E object is a *sdons “CreativeWork”*, it might be required to construct custom objects that can represent the C2E more than just a *sdons “CreativeWork”*. C2E has concept of '*Content Types*' that are used to construct the object models that are used in the C2E implementations.

Content Types are defined in C2E’s object’s *“c2eContainer”* attribute.

Example

```
"c2eContainer": [
    {
      "@id": "c2ens:c2eContentTypes",
      "@type": "sdons:Collection",
      "c2eContentTypes" : [
        {
          "@id": "c2ens:c2eid-xxx/content-type/SampleContentType",
          "@type": "sdons:CreativeWork",
          "Property1": {"@type":"xsd:string"},
          "Property2": {"@type":"xsd:integer"}
        },
        {
          "@id": "c2ens:c2eid-xxx/content-type/SampleContentType",
          "@type": "sdons:CreativeWork",
          "Property1": {"@type":"xsd:string"},
          "Property2": {"@type":"xsd:integer"}
        }
      ]
    }
  ]
```

5.1 Define mandatory attributes

In Example mentioned above, object to define *Content Types* is constructed with following three necessary attributes `"@id"` `"@type"` and `"c2eContentTypes"`.

`"c2eContentTypes"` attribute would be defined in `"@context"` as `"c2eContentTypes": "@graph"`.

5.2 Define C2E Content Types

List of *Content Types* is define using `"c2eContentTypes"` attribute as mentioned in above example.

5.3 Define C2E Content Type Object

```
{
  "@id": "c2ens:c2eid-xxx/content-type/SampleContentType",
  "@type": "Your Content Type",
  "Property1": {"@type":"xsd:string"},
  "Property2": {"@type":"xsd:integer"}
}
```

`"@id"` is defined as following pattern `"c2ens:c2eid-xxx/content-type"`.

`"@type"` should be valid compact IRI link to define the object type.

5.3.1 Define C2E Content Type Object Attributes

First define attributes for example `"Property1"` or `"Property2"` as an valid compact IRI links in `"@context"`. For example:

```
"@context" {
  .....................
  .....................
  "Property1": "c2eTerm:c2eid-xxx/content-type/attribute/Property1"
  "Property2": "c2eTerm:c2eid-xxx/content-type/attribute/Property2"
  .....................
  .....................
}
```

Make sure you should not define Content Type attribute more than once.

After defining attributes in `"@context"` now you can define these into the Content Type.

5.3.2 Define Data Type for C2E Content Type Object Attribute

It is required to define data type of Content Type Object attribute. Data type could be base on xsd or other data type standards. For example:

`"Property1": {"@type":"xsd:string"}`

It will make Content Type Object like:

```
{
  "@id": "c2ens:c2eid-xxx/content-type/SampleContentType",
  "@type": "Your Content Type",
  "Property1": {"@type":"xsd:string"},
  "Property2": {"@type":"xsd:integer"}
}
```

Following JSON-LD specification link has mention namespaces which has valid data types to be used.

JSON-LD 1.1

5.3.3 Define Relationship Between C2E Content Type Objects

To setup relationship between objects. It will required to introduce following relationship types in the `"@context"`.

```
"@context" {
  .....................
  .....................
  "hasOne": "c2eTerm:hasOne",
  "hasMany": "c2eTerm:hasMany",
  "belongsTo": "c2eTerm:belongsTo"
  .....................
  .....................
}
```

And then you can define relationship between the objects as required. For example we can take Content Type objects from example mentioned above and make `"hasOne"` to `"hasMany"` relationship as:

```
"c2eContainer": [
    {
      "@id": "c2ens:c2eContentTypes",
      "@type": "sdons:Collection",
      "c2eContentTypes" : [
        {
          "@id": "c2ens:c2eid-xxx/content-type/student",
          "@type": "sdons:CreativeWork",
          "Property1": {"@type":"xsd:string"},
          "Property2": {"@type":"xsd:integer"},
          "hasMany": "c2ens:c2eid-xxx/content-type/courses"
        },
        {
          "@id": "c2ens:c2eid-xxx/content-type/courses",
          "@type": "sdons:CreativeWork",
          "Property1": {"@type":"xsd:string"},
          "Property2": {"@type":"xsd:integer"},
          "belongsTo": "c2ens:c2eid-xxx/content-type/student"
        }
      ]
    }
  ]
```

These relationship attributes would required `"@id"` of related Content Type Objects.

6. C2E Contents

C2E contents are defined in `c2eContainer` as `c2eContentCatalog` which has Url to `/content/contents.js` path which list all the C2E content files.

```
"c2eContainer": [
    {
      "@id": "c2ens:c2eContentCatalog",
      "@type": "sdons:DataCatalog",
      "c2eContentCatalog": {
              "@id": "c2ens:c2eid-12345/content/contents.json",
              "@type": "sdons:DigitalDocument",
              "url": "/content/contents.json",
              "fileFormate": "application/json"
      }
    }
  ]
```

In `/content/contents.js` file content files would be listed as:

```
{
        "@id": "c2ens:c2eContents",
        "@type": "sdons:Collection",
        "c2eContents": [
                {
                        "@id": "c2ens:c2eid-12345/content/my-project-1.json",
                        "@type": "sdons:DigitalDocument",
                        "@index": "1",
                        "url": "/content/my-project-1.json",
                        "fileFormate": "application/json",
                        "learningResourceType": "Project"
                },
                {
                        "@id": "c2ens:c2eid-12345/content/my-project-2.json",
                        "@type": "sdons:DigitalDocument",
                        "@index": "2",
                        "url": "/content/my-project-2.json",
                        "fileFormate": "application/json",
                        "learningResourceType": "Project"
                }
        ]
}
```

7. Define C2E Source Code

It is required to define source code path of C2E so that it can provide entry point to execute.

C2E source code defined in C2E’s object’s *“c2eContainer”* attribute.

first define `"c2eSource"` in the context as:

```
"@context" {
  .....................
  .....................
  "c2eSourceCode": "@graph"
  .....................
  .....................
}
```

```
"c2eContainer": [
    {
      "@id": "c2ens:c2eSourceCode",
      "@type": "sdons:Collection",
      "c2eSourceCode" : [
        {
          "@id": "c2ens:c2eid-xxx/source/index.js",
          "@type": "sdons:SoftwareSourceCode",
          "url": "/source/index.js",
          "fileFormat": "application/javascript"
        },
        {
          "@id": "c2ens:c2eid-xxx/source/index.html",
          "@type": "sdons:SoftwareSourceCode",
          "url": "/source/index.html",
          "fileFormat": "text/html"
        }
      ]
    }
  ]
```

It should have 2 objects with attribute `"@id"`, having `"c2eTerm:script"` and `"c2eTerm:homepageUrl"`

`"c2eTerm:script"` - It would be script that would be used to execute C2E as code and can be included on HTML document define in `"c2eTerm:homepageUrl"` object.

`"c2eTerm:homepageUrl"` - It is HTML document that will display an C2E document.

8. Define Embeded C2Es

In C2E JSON-LD `c2eContainer` container object, an object is defined with `@id` and `c2eEmbeds` attribute which is used to list all associated C2Es that are required for a C2E.

```
  "c2eContainer": [
    {
      "@id": "c2ens:c2eEmbeds",
      "@type": "sdons:Collection",
      "c2eEmbeds": [
        {
          "@id": "c2ens:c2eid-xxx-1",
          "@type": "C2E",
          "@index": "1",
          "c2eResources": []
        },
        {
          "@id": "c2ens:c2eid-xxx-2",
          "@type": "C2E",
          "@index": "2",
          "subManifest": "/path/to/sample-activity/sample-H5P/c2e.json"
        }
      ]
    }
  ]
```

8.1 Sub-Manifest to define associated C2Es

In case where complex C2Es are needed then a sub-manifest file `c2e.json` can be created on relevant path.

It is defined for objects under `c2eEmbeds` list as:

`"subManifest": "/path/to/sample-activity/sample-H5P/c2e.json"`

Example in 4.4 shows the same sub-manifest declaration.

Note that every `c2e.json` file placed other than root directory would be considered as sub-manifest and should have association with main `c2e.json` manifest file place on the root directory.

9. Licensing

There is `copyright` object with `license` nested object in it which is defined in *4.2.1 Define Metadata* section. This `copyright` object comprise on following JSON-LD objects to fulfill the copyrights and licensing needs.

```
"c2eMetadata": {
    ........
    ........
```

```
  "copyright": {
   "license": {
    "@id": "c2ens:c2eid-xxx/license/id/xxx",
    "@type": "sdons:DigitalDocument",
    "additionalType": "Subscription / Purchase / Open"
    "identifier": "C2e-lsc-xxxxx",
    "dateCreated": "2023-01-01",
    "expires": "2024-01-01",
    "usageInfo": {
     "@type": "sdons:DefinedTermSet",
     "name": "License Terms",
     "hasDefinedTerm": {
      "@type": "sdons:DefinedTerm",
      "name": "Monthly - 1000 views",
      "termCode": "monthly"
     }
    },
    "offers": {
     "@type": "sdons:Offer",
     "price": "19.99",
     "priceCurrency": "USD"
    }
    "licensee": {
      "@id": "c2ens:c2eid-xxxxxx/licensee/id/waqar@curriki.org",
      "@type": "sdons:Person",
      "name": "Waqar",
      "email": "waqar@curriki.org",
      "url": "https://x.com/waqarmnr"
    },
    "encodingFormat": "text/plain or any binary as per need",
    "url": "/path/to/license.txt",
   }
  }
}
```

10. Complete Sample JSON-LD Manifest

Run JSON-LD code in Playground

```
{
 "@context": {
  "c2ens": "https://c2e.curriki.org/",
  "sdons": "https://schema.org/",
  "xsd": "http://www.w3.org/2001/XMLSchema/#",
  "C2E": "sdons:CreativeWork"
  "c2eContainer": {
   "@id": "sdons:Collection",
   "@container": [
    "@graph",
    "@index"
   ]
  },
  "c2eResources": "@graph",
  "c2eContentTypes": "@graph",
  "c2eSourceCode": "@graph",
    "c2eEmbeds": "@graph",
    "c2eTerm": "c2ens:terms/",
    "c2eVendor": {
      "@id": "sdons:vendor",
      "@type": "xsd:string"
    },
    "name": {
      "@id": "sdons:name",
      "@type": "xsd:string"
    },
    "url": {
      "@id": "sdons:url",
      "@type": "xsd:string"
    },
    "fileFormat": {
      "@id": "sdons:fileFormat",
      "@type": "xsd:string"
    },
    "@language": "en",
    "metadata": "c2eTerm:metadata",
    "schemaVersion": {
      "@id": "sdons:schemaVersion",
      "@type": "xsd:string"
    },
    "general": "c2eTerm:mdGeneral",
    "title": {
      "@id": "sdons:headline",
      "@type": "xsd:string"
    },
    "description": {
      "@id": "sdons:description",
      "@type": "xsd:string"
    },
    "keywords": {
      "@id": "sdons:keywords",
      "@type": "xsd:string"
    },
    "author": {
      "@id": "sdons:author",
      "@type": "xsd:string"
    },
    "publisher": {
      "@id": "sdons:publisher",
      "@type": "xsd:string"
    },
    "email": {
      "@id": "sdons:email",
      "@type": "xsd:string"
    },
    "lifecycle": "c2eTerm:mdLifecycle",
    "version": {
      "@id": "sdons:version",
      "@type": "xsd:string"
    },
    "releaseStatus": {
      "@id": "sdons:creativeWorkStatus",
      "@type": "xsd:string"
    },
    "subManifest": {
      "@id": "sdons:url",
      "@type": "xsd:string"
    },
    "copyright": "c2eTerm:copyright",
    "license": {
      "@id": "sdons:license",
      "@type": "xsd:string"
    },
    "encodingFormat": {
      "@id": "sdons:encodingFormat",
      "@type": "xsd:string"
    },
    "copyrightHolder": {
      "@id": "sdons:copyrightHolder",
      "@type": "xsd:string"
    },
    "copyrightNotice": {
      "@id": "sdons:copyrightNotice",
      "@type": "xsd:string"
    },
    "copyrightYear": {
      "@id": "sdons:copyrightYear",
      "@type": "xsd:integer"
    },
    "Property1": "c2ens:c2eid-xxx/content-type/attribute/Property1",
    "Property2": "c2ens:c2eid-xxx/content-type/attribute/Property2"
  },
  "metadata": {
    "@id": "c2ens:c2eid-xxx/metadata",
    "@type": "sdons:Dataset",
    "schemaVersion": "1.0",
    "general": {
      "@id": "c2ens:c2eid-xxx/metadata/general",
      "@type": "sdons:Dataset",
      "title": "My Smaple C2E",
      "description": "My Smaple C2E for Proof of Concept.",
      "keywords": "c2e, e-learning, learning experince"
    },
    "author": {
      "@id": "c2ens:c2eid-xxx/author/id/xxx",
      "@type": "sdons:Person",
      "name": "Waqar Muneer",
      "email": "waqar@curriki.org",
      "url": "https://twitter.com/waqarmnr"
    },
    "publisher": {
      "@id": "c2ens:c2eid-xxx/author/id/xxx",
      "@type": "sdons:Organization",
      "name": "Curriki",
      "email": "info@curriki.org",
      "url": "https://curriki.org"
    },
    "lifecycle": {
      "@id": "c2ens:c2eid-xxx/metadata/lifecycle",
      "@type": "sdons:Code",
      "version": "1.10",
      "releaseStatus": "Beta"
    },
    "copyright": {
      "@id": "c2ens:c2eid-xxx/metadata/copyright",
      "@type": "sdons:Dataset",
      "license": {
        "@id": "c2ens:c2eid-xxx/license/id/xxx",
        "@type": "sdons:DigitalDocument",
        "encodingFormat": "application/text or any binary as per need",
        "url": "/path/to/license.txt"
      },
      "copyrightHolder": {
        "@id": "c2ens:c2eid-xxx/copyrightHolder/id/xxx",
        "@type": "sdons:Person",
        "name": "Waqar Muneer",
        "email": "waqar@curriki.org",
        "url": "https://twitter.com/waqarmnr"
      },
      "copyrightNotice": "This C2E has all rights to Waqar Muneer",
      "copyrightYear": "2023"
    }
  },
  "@id": "c2ens:c2eid-xxx",
  "@type": "C2E",
  "c2eContainer": [
    {
      "@id": "c2ens:c2eResources",
      "@type": "sdons:Collection",
      "c2eResources": [
        {
          "@id": "c2ens:c2eid-xxx/resource/1",
          "@type": "sdons:DigitalDocument",
          "url": "/path/to/file-1.json",
          "fileFormat": "application/json"
        },
        {
          "@id": "c2ens:c2eid-xxx/resource/2",
          "@type": "sdons:DigitalDocument",
          "url": "/path/to/readme.txt",
          "fileFormat": "application/text"
        }
      ]
    },
    {
      "@id": "c2ens:c2eContentTypes",
      "@type": "sdons:Collection",
      "c2eContentTypes" : [
        {
          "@id": "c2ens:c2eid-xxx/content-type/student",
          "@type": "sdons:CreativeWork",
          "Property1": {"@type":"xsd:string"},
          "Property2": {"@type":"xsd:integer"},
          "hasMany": "c2ens:c2eid-xxx/content-type/courses"
        },
        {
          "@id": "c2ens:c2eid-xxx/content-type/courses",
          "@type": "sdons:CreativeWork",
          "Property1": {"@type":"xsd:string"},
          "Property2": {"@type":"xsd:integer"},
          "belongsTo": "c2ens:c2eid-xxx/content-type/student"
        }
      ]
    },
    {
      "@id": "c2ens:c2eSourceCode",
      "@type": "sdons:Collection",
      "c2eSourceCode" : [
        {
          "@id": "c2ens:c2eid-xxx/source/index.js",
          "@type": "sdons:SoftwareSourceCode",
          "url": "/source/index.js",
          "fileFormat": "application/javascript"
        },
        {
          "@id": "c2ens:c2eid-xxx/source/index.html",
          "@type": "sdons:SoftwareSourceCode",
          "url": "/source/index.html",
          "fileFormat": "text/html"
        }
      ]
    },
    {
      "@id": "c2ens:c2eEmbeds",
      "@type": "sdons:Collection",
      "c2eEmbeds": [
        {
          "@id": "c2ens:c2eid-xxx-1",
          "@type": "C2E",
          "@index": "1",
          "c2eResources": []
        },
        {
          "@id": "c2ens:c2eid-xxx-2",
          "@type": "C2E",
          "@index": "2",
          "subManifest": "/path/to/sample-activity/sample-H5P/c2e.json"
        }
      ]
    }
  ]
}
```

Presentation

Extending C2E

Defining C2E Types

Extending C2E … ecs.pdf
15 Feb 2023, 12:42 PM

11. xAPI Statement (C2E Analytics)

Following sample xAPI statement with "verb" as “consumed” shows the access of a C2E.

```
{
    "id": "3564517a-74ba-4a81-976d-6e80497da960",
    "timestamp": "2023-08-22T05:26:40.575Z",
    "actor": {
        "objectType": "Agent",
        "mbox": "mailto:waqar@curriki.org",
        "name": "Waqar"
    },
    "verb": {
        "id": "http://activitystrea.ms/schema/1.0/consume",
        "display": {
            "en-US": "consumed"
        }
    },
    "object": {
        "id": "https://c2e.curriki.org/c2eid-7ffce150-404e-11ee-9cff-c161055bb056",
        "objectType": "Activity",
        "definition": {
            "name": {
                "en-US": "Unit 5: Supply and Demand"
            },
            "extensions": {
                "https://c2e.curriki.org/xAPI/c2eResource": {
                    "@id": "https://c2e.curriki.org/c2eid-7ffce150-404e-11ee-9cff-c161055bb056/resources/Ap05-e8
                    "@type": "https://schema.org/DigitalDocument",
                    "https://schema.org/identifier": {
                        "@type": "https://schema.org/PropertyValue",
                        "https://schema.org/propertyID": "ISBN",
                        "https://schema.org/value": "777777777777777777"
                    },
                    "https://schema.org/fileFormat": "epub",
                    "https://schema.org/url": "/resources/Ap05-e8b33700-4029-11ee-9a4c-6b96eb672531.epub"
                },
                "https://c2e.curriki.org/xAPI/c2eSubjectOf": {
                    "@id": "https://c2e.curriki.org/c2eid-7ffce150-404e-11ee-9cff-c161055bb056/metadata/subjectO
                    "@type": "https://schema.org/CreativeWork",
                    "name": "Anatomy and Physiology All-in-One For Dummies®"
                }
            }
        }
    },
    "stored": "2023-08-22T05:26:41.0682Z",
    "authority": {
        "objectType": "Agent",
        "account": {
            "homePage": "http://trax.test",
            "name": "traxlrs"
        }
    },
    "version": "1.0.0"
}
```

Check following documentation links to know about xAPIs,.

GitHub - adlnet/xAPI-Spec: The xAPI Specification describes communication about learner activity and experiences between technologies.

https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md#parttwo https://github.com/adlnet/xAPI-Spec/blob/master/xAPI-Data.md#statement-properties