NIEM Training Curriculum Overview

# 1. Overview

A user needing an introduction to what NIEM is, and what it is for, and users at the executive level, should start with [About NIEM](https://niem.gov/https%3A//www.niem.gov/about-niem) and [Getting Started](https://www.niem.gov/getting-started).

A newcomer to NIEM with an interest in technical details of NIEM and the practical use and implementation of the NIEM model should start with [Training](http://niem.github.io/training/) to get up to speed. Training is broken down into classes of users.

Training includes

* [IEPD Developer](http://niem.github.io/training/iepd-developer/) walks through the six steps of the IEPD Lifecycle in detail and includes a [Simple IEPD Tutorial](http://niem.github.io/training/iepd-developer/simple-iepd-tutorial/) to help a user build an initial simple IEPD. More advanced tutorials are planned.
	+ In addition an IEPD developer may find the [IEPD Starter Kit](/reference/iepd/iepd-starter-kit/) useful.
* [IEPD Implementer](http://niem.github.io/training/iepd-implementer/) includes techniques and methods that help to implement NIEM exchanges in operational systems.
* A user interested in standing up a new NIEM domain or an enterprise data model for use with multiple IEPD may be interested in [Domain Modeler](http://niem.github.io/training/domain-modeler/) guidance.

Someone who is already familiar with NIEM and is looking for quick reference material to determine or verify how to do something should look at [Reference](/reference/).

Someone who wants to learn about the use of NIEM JSON should review [JSON](/json/).

See the following sections for a more-detailed breakdown of what is available in NIEM training.

# 2. The NIEM.github.io Hierarchy

The NIEM.github.io site is organized hierarchically, with specialized concepts grouped underneath general concept headings.

## 2.1. Home

Link: <http://niem.github.io>

The root page of the NIEM.github.io website. This is meant to be a launching-off point into the rest of the site.

### 2.1.1. Training

Link: <http://niem.github.io/training/>

Tutorials and guidance for specific users and uses of NIEM.

#### 2.1.1.1. Domain Modeler

Link: <http://niem.github.io/training/domain-modeler/>

Training for developers of NIEM domains, enterprise models, and other models for use with multiple IEPDs.

This section provides links and references to other sections relevant to domain modelers.

#### 2.1.1.2. IEPD Developer

Link: <http://niem.github.io/training/iepd-developer/>

Training for people looking to build new information exchanges.

This section provides an overview of the IEPD lifecycle, and a link the simple tutorial and additional resources relevant to IEPD developers.

##### 2.1.1.2.1. Scenario Planning

Link: <http://niem.github.io/training/iepd-developer/scenario-planning/>

Brief overview of artifacts relevant to planning an information exchange

##### 2.1.1.2.2. Analyze Requirements

Link: <http://niem.github.io/training/iepd-developer/analyze-requirements/>

Provides issues relevant to analyzing requirements

##### 2.1.1.2.3. Map and Model

Link: <http://niem.github.io/training/iepd-developer/map-and-model/>

Lays out the process of mapping from source data requirements to NIEM data model objects.

Provides:

* Describes the NIEM mapping document, which can be used to identify source data requirements; this is the data that needs to be conveyed, and how that data is identified in the source data model.
* Provides a link to a sample mapping document.
* Provides a method for linking the source data requirements in terms of NIEM, by searching for concepts in the NIEM Subset Schema Generation Tool (SSGT).
* Provides mapping values, which evaluate the semantic relationship between NIEM terms and source data terms.
* Provides and example of recording NIEM components as a subset in the SSGT.
* Provides examples of how to find terms when exact matches are not obvious.
* Provides considerations about reuse through has-a and is-a relationships.

##### 2.1.1.2.4. Build and Validate

Link: <http://niem.github.io/training/iepd-developer/build-and-validate/>

Provides options for generating schemas, code lists, and describes conformance checking.

##### 2.1.1.2.5. Assemble and Document

Link: <http://niem.github.io/training/iepd-developer/assemble-and-document/>

Describes files that need to be included in an IEPD, IEPD structure, and review of the IEPD.

##### 2.1.1.2.6. Publish and Implement

Link: <http://niem.github.io/training/iepd-developer/publish-and-implement/>

Identifies where to publish your IEPD, along with privacy considerations.

##### 2.1.1.2.7. Simple IEPD Tutorial

Link: <http://niem.github.io/training/iepd-developer/simple-iepd-tutorial/>

Provides an end-to-end walkthrough of developing a very simple IEPD.

This includes:

* Scenario planning
	+ Information flow diagram
* Analyze requirements
	+ Source data elements, classes, code sets, business rules
* Map and model requirements
	+ Use of the mapping document
	+ Use of the SSGT
	+ Generating a subset schema
* Build and validate
	+ IEPD structure
	+ Instance document
* Assemble and document
	+ IEPD structure
	+ Artifacts (files) in an IEPD
* Publish and implement: Github, community registries
* Additional resources

#### 2.1.1.3. IEPD Implementer

Link: <http://niem.github.io/training/iepd-implementer/>

Guidance for people building systems that use NIEM exchanges.

Information includes:

* Use of IEPD Java Bindings
* Information about IEPDs, including additional resources and an IEPD starter kit
* A link to the NIEM tools catalog for additional tools

##### 2.1.1.3.1. Implementation Cookbook

Link: <http://niem.github.io/training/iepd-implementer/implementation-cookbook/>

Techniques for improving implementation of systems that use NIEM exchanges

Techniques include

* Avoid replicating data in a NIEM XML document
	+ Through the use of structures:id and structures:ref
	+ Through use of NIEM metadata
	+ Through use of augmentations
* How to browse content of NIEM
* Enterprise Integration Patterns
	+ Use of standard objects for information processing and message passing
* Reducing size of messages
	+ Server output filter: Reduce size of messages by compressing data as it is sent
	+ Fast Infoset: Reduce size of messages through use of schema-independent message packing
	+ Reduce the size of data through use of EXI (Efficient XML Interchange), a schema-aware message packing and compression standard.

### 2.1.2. Reference

Link: <http://niem.github.io/reference/>

Reference documentation for NIEM artifacts, features, and methods.

#### 2.1.2.1. Information Exchange Package Documentation

Link: <http://niem.github.io/reference/iepd/>

Overview of structure of Information Exchange Package Documentations (IEPDs)

##### 2.1.2.1.1. IEPD Lifecycle

Includes:

* Why an IEPD?
* IEPD Lifecycle
* IEPD Structure
* IEPD Starter Kit

#### 2.1.2.2. NIEM Conformance

Link: <http://niem.github.io/reference/conformance/>

Definition of NIEM conformance, and how to make your artifacts conformant

#### 2.1.2.3. Conformant Artifacts

Link: <http://niem.github.io/reference/artifacts/>

Overview of the kinds of files (and collections of files) used in NIEM

Includes:

* Schema documents:
	+ Reference schema documents
	+ Extension schema documents
	+ Subset Schema Documents
* Sets of schema documents
* Code Lists
* IEPDs
* IEPs

#### 2.1.2.4. Model Concepts

Link: <http://niem.github.io/reference/concepts/>

Basics of how to build data models using the building blocks of XML Schema

Includes:

* Namespaces
* Properties (elements and attributes)
* Types
* Codes & facets
* Adapters
* Associations
* Augmentations
* Local terminology
* Metadata
* References
* Representation pattern
* Roles

#### 2.1.2.5. Content

Link: <http://niem.github.io/reference/content/>

An overview of the content in the NIEM data model

A survey of major objects in the major namespaces in NIEM.

Includes:

* [Common terms](http://niem.github.io/reference/content/common-terms/)
* [Code lists](http://niem.github.io/reference/content/codes/)

#### 2.1.2.6. Releases

Link: <http://niem.github.io/reference/releases/>

An overview of NIEM releases, processes, and the release cycle

Includes:

* References to NIEM governance and the current release
* Description of kinds of releases: major and minor releases
* The NIEM release cycle
* Feedback on releases
* Development of NIEM releases and harmonization
* NIEM core supplements
* Locations where NIEM releases are available
* Release artifacts: files within a NIEM release
* History of NIEM releases
* Statistics of what's in NIEM

#### 2.1.2.7. Tools

Link: <http://niem.github.io/reference/tools/>

An introduction to some tools that can be used with NIEM

Includes:

* SSGT - Schema Subset Generation Tool
* ConTesA - Conformance Testing Assistant
* Movement
* Migration Tool
* The NIEM Tools Catalog

#### 2.1.2.8. Specifications

Link: <http://niem.github.io/reference/specifications/>

Documentation of NIEM specifications and their uses.

##### 2.1.2.8.1. Code Lists

Link: <http://niem.github.io/reference/specifications/code-lists/>

Documentation of the use of code lists as CSV or XML files, with versions managed independently of other parts of exchanges.

#### 2.1.2.9. NIEM Domain Modeling

Link: <http://niem.github.io/reference/domain-modeling-guide/>

A review of the creation and maintenance of a NIEM domain

Includes:

* Background: The purpose of NIEM, What the guide covers, Audience
* Technical basics:
	+ NIEM version architecture
	+ Schema document namespaces and versioning
	+ The NIEM release cycle
	+ Domain updates
* Designing and building domain content: A good overview of how to get started modeling data for NIEM, including names of components and definitions.
* Search for NIEM data components: how to find existing content on which to base your definitions
* Mapping to NIEM: why and how to reuse existing content
* Evaluate data component quality: Questions that help to identify good or bad content
* Overview of NIEM governance and resources
* Locations of NIEM information on the internet
* A glossary of NIEM terms

#### 2.1.2.10. Normative Rules

Link: <http://niem.github.io/reference/normative-rules/>

A list of rules for NIEM artifacts, from several specifications

#### 2.1.2.11. Core supplements

Link: <http://niem.github.io/reference/core-supplement/>

What is a core supplement, and how to use them

#### 2.1.2.12. Reference vs. Extension Schema

Link: <http://niem.github.io/reference/ref-vs-ext/>

The difference between NIEM extension and reference schema documents.

### 2.1.3. JSON

Link: <http://niem.github.io/json/>

A landing page for NIEM-JSON; includes reference and training content.

#### 2.1.3.1. A NIEM-JSON Tutorial

Link: <http://niem.github.io/json/tutorial/>

A front-to-back walkthrough covering the use of NIEM to define JSON messages

This includes:

* Defining data requirements
* Defining a NIEM exchange model
* Defining a JSON-LD context
* Constructing a sample JSON FILE
	+ USING JSON-LD features as needed
	+ A complete sample instance
* Constructing a JSON schema

#### 2.1.3.2. NIEM and JSON Frequently Asked Questions

Link: <http://niem.github.io/json/faq/>

Clarification of a few topics related to NIEM JSON

Includes:

* What is NIEM
	+ Reusable definitions and the NIEM data model
	+ Data resource specifications
	+ NIEM technical specifications
	+ NIEM governance
	+ Different ways to use and participate in NIEM
* Why use NIEM with JSON?
* What is NIEM JSON data?

#### 2.1.3.3. A NIEM-JSON Reference Guide

Link: <http://niem.github.io/json/reference/>

Some resources for NIEM JSON

Topics include:

* Names for JSON data
	+ Reusing NIEM names that are reused across a community
* JSON-LD
	+ Names represent URIs
	+ JSON-LD uses @id for object identification
* NIEM Content Models
	+ Instances and Schemas
* JSON Schema
	+ Some features of JSON Schema that may assist validation of JSON data
* Advanced NIEM Concepts
	+ Use of NIEM JSON in relation to associations, augmentations, metadata, and roles.
* Guidance
	+ A larger drill-down into NIEM-JSON, which covers advanced features of NIEM and how they map to NIEM JSON.