"""

**class BookPublished**

standard format: author(s) (year). title. location: publisher.

initialize: BookPublished(str original)

get\_category: return Defs.BookPublished

get\_original: return str original

is\_parsed: return True if successfully parsed, False if any field fails

get\_authors: return a list of Author / str "error in author(s)"

get\_year: return str year / "error in year"

get\_title: return str book title / "error in title"

get\_location: return str location / "error in location"

get\_publisher: return str publisher / "error in publisher"

get\_source: return str source if present, "" otherwise

encode\_authors(): return Handler.authors

encode\_year(): return Handler.general

encode\_title(): return Handler.title

encode\_location(): return Handler.general

encode\_publisher(): return Handler.general

encode\_source(): return Handler.general

encode()

"""

"""

**class BookInPress**

standard format: author(s) (in press). title. source

initialize: BookInPress(str original)

get\_category: return Defs.BookInPress

get\_original: return str original

is\_parsed: return True if successfully parsed, False if any field fails

get\_authors: return a list of Author / str "error in author(s)"

get\_year: return str year / "error in year"

get\_title: return str book title / "error in title"

get\_source: return str source if present, "" otherwise

encode\_authors(): return Handler.authors

encode\_year(): return Handler.general

encode\_title(): return Handler.title

encode\_source(): return Handler.general

encode()

"""

"""

**class ChapterPublished**

standard format: author(s) (year). chapter title. In editors (Ed./Eds.),

book title ((page addition, )pp. page-page). location: publisher

initialize: ChapterPublished(str original)

get\_category: return Defs.ChapterPublished

get\_original: return str original

is\_parsed: return True if successfully parsed, False if any field fails

get\_authors: return a list of Author / str "error in author(s)"

get\_year: return str year / "error in year"

get\_chapter\_title: return str chapter title / "error in chapter title"

get\_editors: return a list of Author / str "error in editor(s)"

get\_book\_title: return str book title / "error in book title"

get\_start\_page: return str start page / "error in starting page"

get\_end\_page: return str end page / "error in ending page"

get\_page\_addition: return str page addition if present, "" otherwise

get\_location: return str location / "error in location"

get\_publisher: return str publisher / "error in publisher"

get\_source: return str source if present, "" otherwise

encode\_authors(): return Handler.authors

encode\_year(): return Handler.general

encode\_chapter\_title(): return Handler.title

encode\_editors(): return Handler.editors

encode\_book\_title(): return Handler.title

encode\_pages(): return Handler.pages

encode\_page\_addition(): return Handler.page\_addition

encode\_location(): return Handler.general

encode\_publisher(): return Handler.general

encode\_source(): return Handler.general

encode()

"""

"""

**class ChapterInPress**

standard format: author(s) (in press). chapter title. In editors (Ed./Eds.), book title.

initialize: ChapterInPress(str original)

get\_category: return Defs.ChapterInPress

get\_original: return str original

is\_parsed: return True if successfully parsed, False if any field fails

get\_authors: return a list of Author / str "error in author(s)"

get\_year: return str year / "error in year"

get\_chapter\_title: return str chapter title / "error in chapter title"

get\_editors: return a list of Author / str "error in editor(s)"

get\_book\_title: return str book title / "error in book title"

get\_source: return str source if present, "" otherwise

encode\_authors(): return Handler.authors

encode\_year(): return Handler.general

encode\_chapter\_title(): return Handler.title

encode\_editors(): return Handler.editors

encode\_book\_title(): return Handler.title

encode\_source(): return Handler.general

encode()

"""

"""

**class JournalPublished**

standard format: author(s) (year). title. journal, volume(issue), page-page.

initialize: JournalPublished(str original)

get\_category: return Defs.JournalPublished

get\_original: return str original

is\_parsed: return True if successfully parsed, False if any field fails

get\_authors: return a list of Author / str "error in author(s)"

get\_year: return str year / "error in year"

get\_title: return str title("." excluded, "?" & "!" included) / "error in title"

get\_journal: return str journal / "error in journal"

get\_volume: return str volume number / "error in volume number"

get\_issue: return str issue number if present, "" otherwise

get\_start\_page: return str start page / "error in starting page"

get\_end\_page: return str end page / "error in ending page"

get\_source: return str source if present, "" otherwise

encode\_authors(): return Handler.authors

encode\_year(): return Handler.general

encode\_title(): return Handler.title

encode\_journal(): return Handler.general

encode\_volume(): return Handler.general

encode\_issue(): return Handler.issue

encode\_pages(): return Handler.pages

encode\_source(): return Handler.general

encode()

"""

"""

**class JournalInPress**

standard format: author(s) (in press). title. journal. source

initialize: JournalInPress(str original)

get\_category: return Defs.JournalInPress

get\_original: return str original

is\_parsed: return True if successfully parsed, False if any field fails

get\_authors: return a list of Author / str "error in author(s)"

get\_year: return str year / "error in year"

get\_title: return str title("." excluded, "?" & "!" included) / "error in title"

get\_journal: return str journal / "error in journal"

get\_source: return str source if present, "" otherwise

encode\_authors(): return Handler.authors

encode\_year(): return Handler.general

encode\_title(): return Handler.title

encode\_journal(): return Handler.general

encode\_source(): return Handler.general

encode()

"""

"""

**class OnLineJournal**

standard format: author(s) (year). title. journal. Retrieved from source

initialize: OnLineJournal(str original)

get\_category: return Defs.OnLineJournal

get\_original: return str original

is\_parsed: return True if successfully parsed, False if any field fails

get\_authors: return a list of Author / str "error in author(s)"

get\_year: return str year / "error in year"

get\_title: return str title("." excluded, "?" & "!" included) / "error in title"

get\_journal: return str journal / "error in journal"

get\_source: return str source if present, "" otherwise

encode\_authors(): return Handler.authors

encode\_year(): return Handler.general

encode\_title(): return Handler.title

encode\_journal(): return Handler.general

encode\_source(): return Handler.general

encode()

"""

"""

**class Website**

standard format: author(s) (year). title. Retrieved from source

initialize: Website(str original)

get\_category: return Defs.Website

get\_original: return str original

is\_parsed: return True if successfully parsed, False if any field fails

get\_authors: return a list of Author / str "error in author(s)" / str original

get\_year: return str year / "error in year"

get\_title: return str title("." excluded, "?" & "!" included) / "error in title"

get\_source: return str source if present, "" otherwise

encode\_authors(): return Handler.authors

encode\_year(): return Handler.general

encode\_title(): return Handler.title

encode\_source(): return Handler.general

encode()

"""

"""

**class Other**

Reference that cannot be categorized

initialize: Other(str original)

is\_parsed: return False

get\_category: return Defs.Other

get\_original: return original

encode\_original(): return Handler.general

encode()

"""

Handler.general:

head, tail, bold, italic, end

Handler.authors:

ln\_end, char\_end, char\_int, fn\_end, n\_int, seq1, seq2, two\_int, front\_int, back\_int, head, tail, bold, italic, end

Handler.editors:

ln\_end, char\_end, char\_int, fn\_end, n\_int, seq1, seq2, two\_int, front\_int, back\_int, head, tail, bold, italic, end, ed1, ed2

Handler.issue:

head, tail, bold, italic, end, required

Handler.pages:

page\_int, head, tail, bold, italic, end

Handler.page\_addition:

head, tail, bold, italic, end

Handler.title:

ed\_pattern, head, tail, bold, italic, inner\_end, end