



Lionbridge Content API

Development Guidelines

March 27, 2020

Review these guidelines to ensure that the first version of your Connector will:

- have the features required for a production environment
- meet the Lionbridge Connector Certification requirements

After you develop your Connector, the Lionbridge Connector team will use these guidelines as the basis to certify it.

Content API development principles

Over time, the Lionbridge Content API development team releases and deploys new API versions to support bug fixes and enhancements. The Content API development team follows these principles in its ongoing development for minor version upgrades, for example, 1.10.0 to 1.11.0:

- A method may support additional parameters, however these new parameters will be optional. If a new value or a default value is absent, the API will maintain its previous behavior. **All methods are backward compatible.**
- New properties may be added to existing models. These new properties will not change the meaning of existing properties within the model. **All properties are backward compatible.**
- A method can reject any request containing an invalid parameter or invalid header values.

Warning: Malformed requests that were accepted in previous releases, due to loose parsing logic, may result in a bad request response in future releases.

Connector installer

Typically, the content system defines how to install the Connector. Lionbridge recommends the following standards, while respecting the content system's requirements:

Priority	Feature
Requirement	Installer must support installing, uninstalling, and upgrading the Connector. Upgrading should not impact in-progress jobs.
Recommendation	The installer should not require restarting the content system.
Recommendation	Installer should support enabling and disabling the Connector. Most content systems support active and inactive software components.
Recommendation	Installation should query existing version to ensure compatibility before upgrading.

Logging

The Connector logs transaction information, including errors.

Priority	Feature
Requirement	Application must write transaction information to a log file.
Requirement	Application must create a new log file every day.
Recommendation	Application should support multiple logging levels, because Lionbridge Connectors support two logging levels: <ul style="list-style-type: none"> ■ <code>Default</code> (Informational, Warnings, Errors) ■ <code>Debug</code> (verbose logging for all Connector activities)
Recommendation	Application should provide a user interface to locate, consume, and possibly download log files.
Recommendation	Application should clearly indicate that it reports and handles errors encountered while submitting translation jobs.

Payload

The payload quotas ensure that the provisioned server resources can adequately handle workloads. You can increase these quotas on a case-by-case basis. The following table lists quota names and their default values.

Quota	Description	Default Value
MaxJobsPerUser	Maximum number of concurrent submitted jobs per user.	1000
MaxListenersPerSite	Maximum number of listeners in the user's organization.	100
MaxStatusUpdatesPerSite	Maximum number of status updates in the user's organization.	10000

Quota	Description	Default Value
MaxReqsPerJob	Maximum number of requests (content items) per translation job.	1000
MaxSupportAssetsPerJob	Maximum number of support assets per translation job.	1000
MaxFilesPerJob	Maximum number of files per translation job.	1000
N/A	Maximum file size.	1 GB
N/A	Ability to clone a previously delivered job to resubmit it as a new job.	N/A

By default, all organizations are throttled to 10 requests/second. New integrations should either handle the throttling error with a try-catch, or introduce a delay between method invocations.

These quotas are per Lionbridge client, and they are based on the expectation that the client will periodically clean up and delete old jobs. The quotas ensure that the provisioned server resources can adequately handle workloads. If it is likely that you will exceed these quotas, please contact Lionbridge to request an exemption.

XML files within the Payload

Priority	Feature
Requirement	Application must split content into multiple files based on localization requirements. For example, separate SEO content from non-SEO content.
Requirement	Application must present as much context as possible. For example, within the payload XML, Lionbridge should be able to identify the H1, H2, and H3 headings, etc.
Recommendation	Application should provide as much metadata as possible to Lionbridge. Lionbridge can parse out anything not required or that has no value. Construct the payload so that applying an XLST will provide a faithful rendering.

Connector Configuration Options

Typically the Connector has configuration options that an administrator can modify to affect the software performance. For version 1.0 releases, a graphical administration user interface is not required. An administrator can configure these options via a configuration file.

Configuration functionality

Priority	Features
Requirement	Application must support an administrator specifying configuration options via an editable configuration file stored locally on the file system or some other method, to control the application settings described in this section. For the initial release, an administration graphical user interface (GUI) is not required.
Recommendation	Application should support an administrator changing configuration options without restarting the content system.

Configuration options

Priority	Features
Requirement	Application must support an administrator adding new translation providers via an editable configuration file or another interface.
Requirement	Application must support an administrator setting the application logging level to either <code>Default</code> or <code>Debug</code> . The default level is <code>Default</code> . For details, see "Logging" on page 2, above.
Requirement	Application must support configuring which providers are available for users to select. This feature limits user choices to those relevant to a particular application instance, and it prevents users from selecting the wrong provider.
Recommendation	Application should support defining the maximum number of items in an XML file.
Recommendation	Application should support an administrator specifying the following standard proxy implementation settings, where applicable: <ul style="list-style-type: none">■ proxy server address■ port■ protocol

Language Mapping

Priority	Features
Requirement	Application must enable users to map content-system languages to language codes accepted by Lionbridge. For a list of language codes, see "Supported language codes" on page 9.

Application Versioning

Keeping the application and database versions synchronized is a critical component of normal Connector operations. It also facilitates support and troubleshooting.

Priority	Features
Requirement	The Connector must write its version information to the log files on instantiation.
Requirement	The Connector must write relevant configuration information (such API keys and metadata) to the log files on instantiation.
Requirement	The Connector must include the <code>ConnectorName</code> and <code>ConnectorVersion</code> metadata fields in parameters in the endpoints for creating (POST) and modifying (PUT) jobs. They will also be returned in responses that return job details. These metadata fields support troubleshooting. Before certification, please obtain these value from Lionbridge Connectors Support, which you can contact by email at connectors@lionbridge.com .

User Interfaces

The following user interfaces are required in all Connectors. The user interfaces should respect the design and operational considerations of the content system.

Priority	Interface	Description
Requirement	individual item selection	Users can select a single item and send it directly to translation or to the Translation Queue.
Requirement	job creation	Application must restrict the maximum string limit of the <code>jobName</code> and <code>requestName</code> to 250 bytes each.
Requirement	Translation Queue	Users must approve items in the Translation Queue to send them out for translation.
Requirement	Bulk Translation	Users can select and send multiple items for translation in a few steps. This is often a wizard.
Requirement	Translation Status	Users can monitor the translation status of jobs and individual items sent for translation.

Basic Functionality

The Connector must include the following functionality across the UIs described above. Each content system may implement these features differently.

Priority	Features
Requirement	The user can specify the source language for a translation job.
Requirement	The user can specify the target language for a translation job.
Requirement	The user can specify multiple target languages for a translation job.
Requirement	The user can search for content items by string.

Priority	Features
Requirement	The user can search for content items in the content tree, including child items.
Requirement	The user can select one, multiple, or all items from a result set to send for translation.
Requirement	The user can filter results sets by criteria, including strings and dates.
Requirement	The user can send items to Translation Queue.
Requirement	The user can send items immediately for translation.
Requirement	The user can define job metadata: The standard base set includes: job name, job description, purchase order.
Recommendation	The user can define additional job metadata, including: delivery date, SEO required, special instructions, analysis codes, etc.
Requirement	The user can check job status via the Translation Status user interface.
Requirement	The application must check for and accept the <i>redelivery</i> of a job.
Recommendation	Before a redelivery, the application should check and warn for updated content. This reduces the risk of a redelivery overwriting content that has been updated by a user.
Requirement	The submission structure must be in accordance with the defined guidelines in key concepts and Connector workflows .
Requirement	<p>The application must support users checking for job status changes. Use listeners so that the Content API pushes updates. If you use polling, use <code>StatusUpdate</code> for each job endpoint. In addition, the application should have an auxiliary way to query and retrieve translated content in case of malfunctioning listener endpoints or status update expiration.</p> <p>Notes: Use only one listener of each type. If the user created a listener, the Content API implicitly acknowledges <code>StatusUpdate</code> of user jobs before the listener request is delivered. For example, if the listener endpoint is down, the Content API acknowledges <code>StatusUpdate</code>, and it is not included in a query.</p>
Requirement	The application must not allow users to cancel or delete jobs that are in progress.
Requirement	The application must respond correctly to the cancellation status in the API.
Recommendation	The application should support a multi-workflow model.
Recommendation	The application should support including <i>reference material</i> with the payload. Alternatively, you can pass information (instructions) via the description field in Content API. This is text input, restricted to 500 characters downstream. The application should support including additional information as a reference file.

Priority	Features
Recommendation	The application should support users requesting a quote for a job.
Recommendation	The application should hiding completed jobs from the user interface a after configurable time period, for example, a specified number of days.
Recommendation	<p>The application should support users approving requests in jobs, if they are satisfied with the translated content, or rejecting requests in jobs, to reject the translated content for either quality or technical reasons.</p> <ul style="list-style-type: none"> ■ For approving requests in jobs, use the PUT /jobs/{jobId}/requests/approve method. ■ For rejecting requests in jobs, use the PUT /jobs/{jobId}/requests/reject method.

Notifications

The Connector uses notifications to send job information to administrators and end users.

Priority	Features
Requirement	The application must send notifications when it sends out a job for translation.
Requirement	The application must send notifications when a job returns from translation.
Recommendation	An administrator should define email-server information, including hostname, credentials and protocols to use.
Recommendation	Extend the default feature set. Additional features would include notifications when jobs fail, etc.

Team Profiles

Team profiles support separating groups of users, their content, and their jobs.

Priority	Features
Requirement	The application must support assigning specific providers to specific environments or user groups.

Local TM implementation

Priority	Features
Nice to have	This feature varies depending on the content system. The goal is to store translated content locally, then compare translation requests, or compare field-level time stamps, if that functionality exists. The goal is to reduce the payload, so that it is only the delta between deliveries.

Polling

The Content API uses push notifications, which supports polling.

Priority	Features
Recommendation	<p>The Connector can poll the Content API for translated content that the translation provider delivers. Polling is useful in the following scenarios:</p> <ul style="list-style-type: none">■ while waiting for urgent translation jobs■ if there is a short SLA period■ if there are high volumes of content and translated content is delivered for one target language at a time <p>When using polling, use a default frequency of 60 minutes. If required, you can configure a shorter polling frequency. However, for large file payloads, a longer polling frequency is recommended.</p>

Passwords

Priority	Features
Requirement	Passwords must be hashed and not stored in cleartext.

Best practices in Connector development

Best Practice	Explanation
Implement a catch-and-retry for 50x or 429 (throttling) errors from the API, with an exponential back-off retry interval. This increases the reliability of the integration.	Occasionally, the API can experience load spikes that require additional system resources. The back end is designed to scale up when these are detected. However, there may be a delay before these capacities are available. There is also a rate throttling limit for API requests that users can hit when making a lot of API requests in a short amount of time.
Introduce a slight delay between method invocations when iterating over a large number of jobs or requests. This smooths out the load spikes.	
Set the <code>Accept</code> request-header field to the expected response content type, when retrieving the translated file from the API. For most use cases, setting the field to <code>application/octet-stream</code> will suffice.	This prevents the situation of receiving a base64 string instead of the expected translated file.
Enable mashing payload files together.	This avoids hundreds of files in a single payload file.
Retrieve at the job level (as opposed to the file level).	
Use the <code>LL-CC</code> format for the language codes, listed in "Supported language codes" on page 9.	

Best Practice	Explanation
Validate your language list.	This prevents a situation where you programmed <code>pt-br</code> , but you intended to program <code>pt-pt</code> .
Create as much business logic within the Connector as possible.	For example, with job approval, ensure that after delivery the translated content does not become publicly available until after a user reviews and accepts it. This can be configurable by the user.
Connectors should include a User Guide and an Installation Guide for effective support post deployment.	Documentation enables users to be self-sufficient.

What's Next?

If you have any questions about the process or the best practices described in this document...	If you have any questions during the development process...	After you have developed your Connector...
Please email connectors@lionbridge.com to contact the Lionbridge Connector Team.		
We can set up a call with you.	We can answer your questions.	Request a certification meeting with Lionbridge to certify the Connector for production use. Note: Certification is required before Production access is granted.

Supported language codes

Name	Description
aa-ET	Afar
ab-GE	Abkhaz [Abkhazian]
ach-UG	Acholi [Acoli]
ADH	Adhola (Uganda)
af-ZA	Afrikaans
ak-GH	Akan
alg-CA	Algonquin
am-ET	Amharic

Name	Description
apa-us	Apache
ar-AE	Arabic (United Arab Emirates)
ar-DZ	Arabic (Algeria)
ar-EG	Arabic (Egypt)
ar-IL	Arabic (Israel)
ar-IQ	Arabic (Iraq)
ar-JO	Arabic (Jordan)
ar-KW	Arabic (Kuwait)
AR-LB	Arabic (Lebanon)
ar-LY	Arabic (Libya)
ar-MA	Arabic (Morocco)
arn-CL	Mapudungun
ar-OM	Arabic (Oman)
ar-qa	Arabic (Qatar)
ar-SA	Arabic (Saudi Arabia)
ar-SD	Arabic (Sudan)
ar-TN	Arabic (Tunisia)
ar-xg	Arabic (Gulf)
ar-XN	Arabic (International) [Modern Standard]
as-IN	Assamese
az-AZ	Azerbaijani
az-Latn-AZ	Azerbaijani (Latin)
bal-PK	Balochi
ban-IN	Bali [Balinese]
ba-RU	Bashkir
be-BY	Belarusan [Belarusian]

Name	Description
ber-MA	Tamazight [Berber]
bfi-IN	Bagheli
bg-BG	Bulgarian
bhb-IN	Bhili
bh-IN	Bhojpuri
bih-in	Bihari
bin-NG	Edo [Bini]
bi-VU	Bislama
bn-BD	Bengali [Bangla]
bn-IN	Bengali (India)
bo-CN	Tibetan
br-FR	Breton
brx-IN	Bodo
bs-BA	Bosnian
bs-Cyrl-BA	Bosnian (Cyrillic)
bs-Latn-BA	Bosnian (Latin)
ca-ES	Catalan
ca-ES-V	Valencian
ceb-PH	Cebuano
chk-FM	Chuukese [Trukese]
co-FR	Corsican (France)
co-IT	Corsican (Italy)
cr-CA	Cree
cs-CZ	Czech
cy-GB	Welsh
da-DK	Danish

Name	Description
de-at	German (Austria)
de-BE	German (Belgium)
de-ch	German (Switzerland)
de-de	German (Germany)
de-li	German (Liechtenstein)
de-lu	German (Luxembourg)
dua-CM	Duala
dv-MV	Dhivehi [Maldivian]
dyu-BF	Dyula
dz-BT	Dzongkha [Bhutanese]
ee-GH	Ewe
el-CY	Greek (Cyprus)
el-GR	Greek
en-au	English (Australia)
en-bz	English (Belize)
en-ca	English (Canada)
en-cpe	Creoles and pidgins, English-based (Other)
en-cy	English (Cyprus)
en-EG	English (Egypt)
en-gb	English (United Kingdom)
en-HK	English (Hong Kong)
en-id	English (Indonesia)
en-ie	English (Ireland)
en-IN	English (India)
en-jm	English (Jamaica)
en-JO	English (Jordan)

Name	Description
en-KE	English (Kenya)
en-LB	English (Lebanon)
en-LK	English (Sri Lanka)
en-MT	English (Malta)
en-MW	English (Malawi)
en-MY	English (Malaysia)
en-NG	English (Nigeria)
en-nz	English (New Zealand)
en-PH	English (Philippines)
en-PK	English (Pakistan)
en-SA	English (Saudi Arabia)
en-SG	English (Singapore)
en-us	English (United States)
en-xg	English (Gulf)
en-XN	English (International)
en-za	English (South Africa)
eo-XN	Esperanto
es-ar	Spanish (Argentina)
es-BO	Spanish (Bolivia)
es-CL	Spanish (Chile)
es-CO	Spanish (Colombia)
es-CR	Spanish (Costa Rica)
es-CU	Spanish (Cuba)
es-DO	Spanish (Dominican Republic)
es-EC	Spanish (Ecuador)
es-es	Spanish (Spain)

Name	Description
es-GT	Spanish (Guatemala)
es-HN	Spanish (Honduras)
es-mx	Spanish (Mexico)
es-NI	Spanish (Nicaragua)
es-PA	Spanish (Panama)
es-PE	Spanish (Peru)
es-pr	Spanish (Puerto Rico)
es-PY	Spanish (Paraguay)
es-SV	Spanish (El Salvador)
es-US	Spanish (United States)
es-UY	Spanish (Uruguay)
es-VE	Spanish (Venezuela)
es-XL	Spanish (Latin America)
es-XN	Spanish (International)
et-EE	Estonian
etu-NG	Ejagham
eu-ES	Basque
fa-AF	Dari
fa-in	Farsi [Persian]
fa-IR	Persian [Farsi]
fat-GH	Fanti
fi-FI	Finnish
fil-PH	Filipino (Philippines)
fo-FO	Faroese
fr-011	French (West and Central Africa)
fr-be	French (Belgium)

Name	Description
fr-ca	French (Canada)
fr-CD	French (DR Congo)
fr-ch	French (Switzerland)
fr-CI	French (Ivory Coast)
fr-CM	French (Cameroon)
fr-DZ	French (Algeria)
fr-fr	French (France)
fr-GN	French (Guinea)
fr-lu	French (Luxembourg)
fr-MA	French (Morocco)
fr-QM	French (Africa)
fr-TG	French (Togo)
fr-TN	French (Tunisia)
fr-xn	French (International)
fuf-GN	Pular
fy-NL	Frisian
gaa-GH	Ga
ga-ie	Irish
gd-GB	Gaelic [Scottish]
gil-kir	Gilbertese (Kiribati)
gl-ES	Galician [Gallegan]
gu-IN	Gujarati
gur-GH	Farefare
ha-NE	Hausa
he-IL	Hebrew
hi-FJ	Hindustani (Fiji)

Name	Description
hi-IN	Hindi
hil-PH	Hiligaynon
hmn-LA	Hmong
ho-PG	Hiri Motu (Papua New Guinea)
hr-ba	Croatian (Bosnia and Herzegovina)
hr-HR	Croatian
ht-HT	Creole (Haiti)
hu-HU	Hungarian
hy-AM	Armenian
hz-na	Herero
id-ID	Indonesian [Bahasa]
igb-NG	Ebira [Igbira]
ig-NG	Igbo
ike-CA-NU	Inuktitut (Nunavut)
ike-CA-QC	Inuktitut (Nunavik)
ilo-PH	Ilocano [Iloko]
is-IS	Icelandic
it-CH	Italian (Switzerland)
it-IT	Italian (Italy)
its-NG	Isekiri [Itsekiri]
ja-jp	Japanese
jam-jm	Jamaican Patois (Patwa)
jaz-NC	Jawe
jv-ID	Javanese
ka-GE	Georgian
kam-KE	Kamba

Name	Description
kck-ZW	Kalanga
kg-CD	Koongo [Kikongo]
kik-KE	Kikuyu [Gikuyu]
kk-cn	Kazakh (China)
kk-KZ	Kazakh
kl-GL	Greenlandic [Kalaallisut]
km-KH	Khmer [Cambodian]
kn-IN	Kannada
kok-IN	Konkani
ko-kr	Korean
kri-SL	Krio
ks-IN	Kashmiri
ks-PK	Kashmiri (Pakistan)
ksw-mymr	Karen (S'gaw)
ku-IQ	Kurdish, Central [Sorani]
ku-TR	Kurdish, Northern [Kurmanji]
ky-KG	Kirghiz [Kyrgyz]
la-VA	Latin
lb-LU	Luxembourgish [Letzeburgesch]
lg-UG	Luganda
ln-CG	Lingala
lo-LA	Lao
loz-ZM	Lozi
lt-LT	Lithuanian
lu-CD	Luba-Lulua [Tshiluba]
luo-ke	Luo (Dholuo)

Name	Description
lv-LV	Latvian
mai-IN	Maithili
mg-MG	Malagasy
mi-NZ	Maori
mk-MK	Macedonian
ml-IN	Malayalam
mni-IN	Manipuri [Meitei]
mnk-GM	Mandinka [Mandingo]
mn-MN	Mongolian
moh-CA	Mohawk
mr-IN	Marathi
ms-MY	Malay (Malaysia)
ms-SG	Malay (Singapore)
mt-MT	Maltese
mtq-vn	Muong
myi-IN	Mina (India)
my-MM	Burmese
myn-GT	K'iche' [Quiche]
naq-NA	Nama
nb-NO	Norwegian [Bokmål]
nd-ZW	Ndebele, Northern [Sindebele]
ne-NP	Nepali
nic-GH	Dagbani
nl-BE	Dutch (Belgium)
nl-NL	Dutch (The Netherlands)
nn-NO	Norwegian [Nynorsk]

Name	Description
nr-ZA	Ndebele, Southern
nso-ZA	Sotho, Northern [Pedi, Sepedi]
nv-US	Navajo
ny-MW	Nyanja
oc-ES	Occitan
om-ET	Oromo
or-IN	Orya [Oriya]
pag-PH	Pangasinan
pa-IN	Panjabi [Punjabi]
pam-PH	Pampanga (Philippines)
pap-AW	Papiamentu [Papiamentu]
pa-PK	Punjabi (Pakistan)
pl-PL	Polish
ps-AF	Pashto [Pushto]
pt-AO	Portuguese (Angola)
pt-BR	Portuguese (Brazil)
pt-MZ	Portuguese (Mozambique)
pt-PT	Portuguese (Portugal)
qu-BO	Quechua
qu-pe	Quechua (Peru)
rm-CH	Romansch
roa-IT	Lombard
ro-MD	Romanian (Moldova)
rom-MK	Romani, Vlax (Romania)
ro-RO	Romanian (Romania)
ru-BY	Russian (Belarus)

Name	Description
ru-EE	Russian (Estonia)
ru-IL	Russian (Israel)
ru-kg	Russian (Kyrgyzstan)
ru-KZ	Russian (Kazakhstan)
ru-lt	Russian (Lithuania)
ru-LV	Russian (Latvia)
ru-RU	Russian
ru-UA	Russian (Ukraine)
ru-xn	Russian (International)
rw-RW	Kinyarwanda
sa-IN	Sanskrit
sat-IN	Santali
sco-GB	Scots
sd-PK	Sindhi
se-NO	Saami [Lapp]
sgn-GB	Sign Language, British
shg-BW	Shua
si-LK	Sinhala [Sinhalese]
skr-PK	Saraiki
sk-SK	Slovak
sl-SI	Slovenian [Slovene]
sm-WS	Samoan
sn-ZW	Shona
so-SO	Somali
sq-AL	Albanian
sr-Cyrl-RS	Serbian [Cyrillic]

Name	Description
sr-Latn-ME	Serbian [Latin] (Montenegro)
sr-Latn-RS	Serbian [Latin]
srp-ME	Montenegrin
sso-LS	Sissano
ss-SZ	Swati [Swazi]
st-LS	Sotho, Southern [Sesotho]
su-ID	Sundanese
sus-GN	Susu
sv-FI	Swedish (Finland)
sv-SE	Swedish (Sweden)
sw-KE	Swahili
syl-BD	Sylheti
ta-IN	Tamil (India)
ta-LK	Tamil (Sri Lanka)
tcy-IN	Tulu
te-IN	Telugu
tem-SL	Timne [Themne]
TEO	Teso (Uganda)
tet-TL	Tetum
tg-TJ	Tajik
th-TH	Thai
ti-ER	Tigrinya [Tigrigna]
tig-ER	Tigre
tiv-NG	Tiv
tk-TM	Turkmen
tll-CD	Tetela

Name	Description
tl-PH	Tagalog
tn-BW	Tswana [Setswana]
to-TO	Tongan
tpi-PG	Tok Pisin (Papua New Guinea)
tr-CY	Turkish (Cyprus)
tr-TR	Turkish
ts-ZA	Tsonga
tt-RU	Tatar
tlv-TV	Tuvaluan [Tuvalu]
tw-GH	Twi
ug-CN	Uighur [Uyghur]
ug-ny	Nyankole
uk-UA	Ukrainian
und-und	All Languages
ur-IN	Urdu (India)
ur-PK	Urdu (Pakistan)
uz-Cyrl-UZ	Uzbek (Cyrillic)
uz-Latn-UZ	Uzbek (Latin)
uz-UZ	Uzbek
ve-ZA	Venda
vi-VN	Vietnamese
wo-SN	Wolof
wuu-CN-CN	Wu Chinese [Shanghainese]
xh-ZA	Xhosa
xsm-GH	Kasem
yao-MW	Yao

Name	Description
yi-IL	Yiddish
yo-NG	Yoruba
zh(-Hans)-MY	Chinese (Malaysia)
zh-cn	Chinese [Simplified]
zh-hk	Chinese (Hong Kong)
zh-mo	Chinese (Macau)
zh-SG	Chinese (Singapore)
zh-tw	Chinese [Traditional]
zh-xy	Chinese (Cantonese)
zu-ZA	Zulu