Factsheet

Daten zum Tool

Tool	FromThePage
URI	https://fromthepage.com/
Letzter Zugriff	27.03.2023
Reviewer	Sabrina Strutz

Allgemeines

Scope des Tools	Transkription
Softwareumgebung (Remotesystem im Browser / Lokaler Client)	Browser-Anwendung
Kostenmodell (Kostenübersicht / Open Source)	Kleinprojekte: bis zu 200 Seiten - kostenlos Betreuung von Institutionen: bis zu 50.000 Seiten - ca. 3600 Dollar/Jahr
Accountsystem	✅
Einbindung anderer Systeme	✅ (IIIF)
Erforderte Code Literacy	[sehr gering?]
Interface-Sprachen (ISO 639-1)	en, es, fr, pt, de (in Arbeit)
Wartung und ständige Erweiterung	✅ (solange Institutionen Projekte über FTP abwickeln)
Einbindung der Community	via Github
Dokumentation	[ausführlich?]
Verfügbarkeit von Tutorials	[keine Schritt-für-Schritt-Anleitungen, aber Videomaterial zu Features und Projekten]
Interoperabilität	
Aktiver Support/Community (Forum, Slack, Issue Tracker etc.)	✅ Issues auf Github, Chat und Mailkontakt

Datenupload

Unterstützte Dateiformate	JPG, PNG, GIF, PDF, ZIP, IIIF (Manifest + Collection) Zusätzlich: Import von CONTENTdm, Internet Archive
Informationen zur Datensicherheit	We make no intellectual property claims over images, transcripts, annotations or comments posted on FromThePage.Transcriptions and translations contributed via FromThePage inherit the ownership & copyright attached to the originals. We reserve the right to remove any content you post for any reason. We may keep backup copies of your deleted post or account on our servers for up to 14 days after you delete it.
Zugänglichkeit von verschiedenen Standorten/Geräten	✅
Einschränkungen hinsichtlich der Datenmenge	❌
Verlustfreier Upload von bereits bearbeiteten Dokumenten	[nicht anwendbar bei Bilddateien]
Unterstützung von IIIF-Import	✅

Daten- und Toolverwaltung

Zentrale/dezentrale Verwaltungsmöglichkeit	✅ mehrere Project Owner möglich
Versionskontrolle	✅ im Interface integriert
Projektspezifische Einstellungen	✅
API	✅

User Management

Personenverwaltung	✅Hinzufügen/Einschränkung von Mitarbeitenden möglich
Interne Kommunikationsmöglichkeiten (z. B. Annotationsrichtlinen, Kommentarfunktionen usw.)	✅ Anlegen von eigenen Hilfstexten, Transkriptionsrichtlinien, Anleitung zum Subject Linking, Forum für Projektdiskussionen, Kommentarfunktion für jede Transkription

Datenbearbeitung

Allgemein

Simultanes Arbeiten	❌ Während Bearbeitung der Transkription einer Bilddatei, ist diese gesperrt

Transkriptionstools

Komplexitätsgrad beim Mark-Up - z.B. Verfügbarkeit von Buttons, Drag&Drop-Funktion	Silent Mark-Up Tags & Buttons
Default-Settings entsprechend bestimmten Standard (TEI)	✅ HTML, TEI
Anpassungsmöglichkeit und Validierung entsprechend projektspezifischen Konventionen/Schemata	Subject Linking - Erweiterung um eigene Kategorien
Definition eigener/projektspezifischer Tags	❌
Metadaten-Anreicherung	✅ Allgemeine Projektmetadaten sowie Metadaten zu einzelnen Werken
Layoutmöglichkeiten (z.B. Tabellendarstellung, Spalten wie in Zeitschriften, etc.)	Tabellendarstellung über Markdown, Formularlayout für Werk konfigurierbar

Eigene Indexierung	✅ Taxonomie über Subject Index
Möglichkeit von Textvergleich bzw. Arbeit an Variantenapparat	❌
Unterstützte Zeichenkodierungen (UTF-8?)	[keine Angabe]
Ansichtsmöglichkeiten (Bearbeitungsansicht, Vorschau, Synopsen-Ansicht)	Bearbeitungsansicht, Bild-Editor-Synopse, Vorschau, *Single-Page-Darstellung:* 1:1-Darstellung (Zeilenumbrüche werden beibehalten) *Multi-Page-Darstellung:* nur Thumbnail-Bilder für die Faksimiles, zur Darstellung von Suchergebnissen (Absatzumbrüche werden beibehalten)
Verlinkung von Entitäten, NER	✅ Subject Linking (Personen, Orte und eigene Kategorien)

Datenexport

Unterstützte Dateiformate	HTML, Plain Text (Verbatim, Searchable, Expanded), TEI, IIIF, PDF, DOCX, CSV, ZIP (nur für Projektleitung)
Datenverlust (Erhalt von Annotationen, die bereits vor Verwendung des Tools gemacht wurden?)	[nicht anwendbar?]
Validierungsmöglichkeit für TEI-XML	
Datenaufbewahrung nach Export?	

RIDE Checkpoints

Beispieldatensatz: https://ride.i-d-e.de/issues/issue-15/teitok/factsheet/
Kriterienkatalog: https://www.i-d-e.de/publikationen/weitereschriften/criteria-tools-version-1/

Review als Tool-Nutzerin

General information		
Software type (?)	What type of software is it? (cf. Catalogue 0.1.1)	web-basiert
Identification of the environment (?)	On which platform runs the tool? (cf. Catalogue 1.4)	Webbrowser
Purpose	For what purpose was the tool developed? (cf. Catalogue 1.5)	Transkription
Funding	Which is the financial model of the tool? (cf. Catalogue 1.6)	kostenpflichtig ab 200 Seiten
Maturity	What is the development stage of the tool? (cf. Catalogue 1.5)	zugänglich (released?)
Methods and implementation		
Programming Language	Which programming languages and technologies are used? (cf. Catalogue 2.3)	???
Reuse	Does the tool reuse portions of other existing software? (cf. Catalogue 2.3)	???
Input format	Which input formats are supported? (cf. Catalogue 2.4)	JPG, PNG, GIF, PDF, ZIP, IIIF (Manifest + Collection) Zusätzlich: Import von CONTENTdm, Internet Archive
Output format	Which output formats are supported? (cf. Catalogue 2.4)	HTML, Plain Text (Verbatim, Searchable,

		Expanded), TEI, IIIF, PDF, DOCX, CSV, ZIP (nur für Projektleitung)
Encoding	Which character encoding formats are supported? (cf. Catalogue 2.4)	UTF-8
Encoding preprocessing	Is a pre-processing conversion included?	???
Dependencies	Does the documentation list dependencies on other software, libraries or hardware? (cf. Catalogue 3.2)	❌
Dependencies installation	If yes, is the software handling the installation of dependencies during the general installation process (you don't have to install them manually before the installation)?	[nicht anwendbar]
Documentation and support		
Documentation	Is documentation and/or a manual available? (tool website, wiki, blog, documentation, or tutorial) (cf. Catalogue 3.4)	✅ Documentation (für Projekteigentümer) + Github-Issue-Mechanismus
Documentation format	Which format has the documentation? (cf. Catalogue 3.3)	HTML
Documentation parts	Which of the following sections does the documentation contain? (cf. Catalogue 3.3)	Project Owner Documentation, FAQ, Descriptions of functionalities (Upload, Transcription, User Management, Project Configuration, Subject Linking, Export), verschiedene How-Tos (inkl. Videomaterial), Installationsanleitung

Documentation language	In what languages is the documentation available? (cf. Catalogue 3.3)	Englisch
Support	Is there a method to get active support from the developer(s) or from the community? (cf. Catalogue 3.4)	✅
From of support	Which form of support is offered? (cf. Catalogue 3.4)	E-Mail- & Chat-Support, Telefonische Beratung (für Institutionen) Github Issues, Slack Channel, Chat Option, Google Group (für Bug Fixes und Releases)
Issue tracker	Is it possible to post bugs or issue using issue tracker mechanisms? (cf. Catalogue 3.4)	✅
Usability and sustainability		
Build and install	Grade how straightforward it is to build or install the tool on a supported platform: (cf. Catalogue 3.6)	Very easy
Tests	Is there a test suite, covering the core functionality in order to check that the tool has been correctly built or installed? (cf. Catalogue 3.7)	✅
Portability and interoperability	On which platforms can the tool/software be deployed? (cf. Catalogue 3.8)	Linux & Windows
Devices	On which devices can the tool/software be deployed? (cf. Catalogue 3.8)	Desktop & mobil
Browsers	If the tool is web-based: On which browsers can the tool/software be deployed? (cf. Catalogue 3.8)	Alle Browser

Plugins	If the tool is web-based: Does the tool rely on browser plugins? (cf. Catalogue 3.8)	keine Plug-Ins nötig
API	Is there an API for the tool? (cf. Catalogue 3.8)	✅
Code	Is the source code open? (cf. Catalogue 3.9)	✅
License	Under what license is the tool released? (cf. Catalogue 3.9)	Affero GPL
Credits	Does the software make adequate acknowledgement and credit to the project contributors? (cf. Catalogue 3.9)	✅
Registered	Is the tool/software registered in a software repository? (cf. Catalogue 3.9)	✅ Github
Possible contribution	If yes, can you contribute to the software development via the repository/development platform?	✅
Analysability, extensibility, reusability of the code		
Analysability	Can the code be analyzed easily (is it structured, commented, following standards)? (cf. Catalogue 3.10)	?-?
Extensibility	Can the code be extended easily (because there are contribution mechanisms, attribution for changes and backward compatibility)? (cf. Catalogue 3.10)	✅
Reusability	Can the code be reused easily in other contexts (because there are appropriate interfaces and/or a modular architecture)? (cf. Catalogue 3.10)	?-?
Security and privacy	Does the software provide sufficient information about the treatment of the data entered by the users? (cf. Catalogue 3.11)	✅
Supportability and maintenance	Is there information available whether the tool will be supported currently and in the future? (cf. Catalogue 3.12)	✅?

Citability	Does the tool supply citation guidelines (e.g. using the Citation File Format)? (cf. Catalogue 3.13)	❌
User interaction, GUI and visualization		
User profile	What kind of users are expected? (cf. Catalogue 4.1)	GeWi-Forschungsinstitutionen und Forschende als Tool-Nutzende
User interaction	What kind of user interactions are expected? (cf. Catalogue 4.1)	Projektkonfiguration und -bearbeitung
User Interface	What kind of interface does the tool provide? (cf. Catalogue 4.2 and 0.1.1)	webbasiertes GUI
Visualization	Does the tool provide a particular visualization (in terms of analysis) of the input and/or the output data? (cf. Catalogue 4.3)	✅ (z. B. Projektstati)
User empowerment	Is the user allowed to customize the functioning of the tool and the output configuration? (cf. Catalogue 4.4)	✅
Accessibility	Does the tool provide particular features for improving accessibility, allowing „people with the widest range of characteristics and capabilities” to use it? (cf. Catalogue 4.5)	?-?

FAIR4RS Prinzipien

Quelle: https://zenodo.org/record/6623556#.ZAm3EnbMKUn

F: Software, and its associated metadata, is easy for both humans and machines to find.	
• F1. Software is assigned a globally unique and persistent identifier.	
○ F1.1. Components of the software representing levels of granularity are assigned distinct identifiers.	
○ F1.2. Different versions of the software are assigned distinct identifiers.	
• F2. Software is described with rich metadata.	
• F3. Metadata clearly and explicitly include the identifier of the software they describe.	
• F4. Metadata are FAIR, searchable and indexable.	

A: Software, and its metadata, is retrievable via standardized protocols.	
• A1. Software is retrievable by its identifier using a standardized communications protocol.	
○ A1.1. The protocol is open, free, and universally implementable	
○ A1.2. The protocol allows for an authentication and authorization procedure, where necessary.	
• A2. Metadata are accessible, even when the software is no longer available.	

I: Software interoperates with other software by exchanging data and/or metadata, and/or through interaction via application programming interfaces (APIs), described through standards.	
• I1. Software reads, writes and exchanges data in a way that meets domain-relevant community standards.	
• I2. Software includes qualified references to other objects.	

R: Software is both usable (can be executed) and reusable (can be understood, modified, built upon, or incorporated into other software).	
● R1. Software is described with a plurality of accurate and relevant attributes.	
○ R1.1. Software is given a clear and accessible license.	
○ R1.2. Software is associated with detailed provenance.	
● R2. Software includes qualified references to other software.	
● R3. Software meets domain-relevant community standards.	