

# Core Sprachmodule V2

- [Versionen](#)
- [Autoren](#)
- [Datentypen](#)
- [Aktivitäten](#)
- [Policies](#)
- [Kontextmodule](#)

## Versionen

Datum	Notiz	Version
30.06.2019	Erste Veröffentlichung als eigenständiges Dokument	2.0
19.03.2019	Veröffentlicht als Abschnitt der Compiler Spezifikation	1.0

## Autoren

Name	Institution	Email
Fabian Bruckner	Fraunhofer ISST	<a href="mailto:fabian.bruckner@isst.fraunhofer.de">fabian.bruckner@isst.fraunhofer.de</a>

D° enthält keine fest integrierten Datentypen, Aktivitäten oder Policies. Es liegt in der Verantwortung der Anwender die notwendigen Funktionalitäten für die Sprache zur Verfügung zu stellen, damit die Entwicklung von Applikationen mit D° möglich wird. Um die Verwendung von D° einfacher zu machen und den Aufwand bei der Implementation eigener Sprachelemente zu reduzieren, verfügt D° bei allen erweiterbaren Systemen über ein 'core'-Modul, welches einige Elemente enthält, welche durch Anwender verwendet werden können.

Dabei ist es nicht das Ziel mit den 'core'-Modulen einen maximal großen Funktionsumfang abzubilden. Stattdessen sollen diverse häufig benötigte Funktionen, Datentypen und Policies jedem Anwender zur Verfügung gestellt werden.

Nachfolgend werden die einzelnen Elemente der jeweiligen Module aufgelistet. Alle Sprachelemente, die in diesen Core Modulen definiert werden, können entweder ohne die Angabe eines Namensraums oder mit dem Namensraum `core` verwendet werden.

## Datentypen

Dieser Abschnitt betrachtet die Validatoren und Datentypen, welche im entsprechenden Core Sprachmodul definiert werden.

### Validatoren

	Namespace	Name	Parameter	Beschreibung	Änderung in Version (en)	Hinzugefügt in Version
1	core	VoidValidator	-	Akzeptiert alle Werte.	-	V1.0
2	core	RegexValidator	<code>pattern</code> - Ein Regulärer Ausdruck	Dieser Validator akzeptiert alle Werte, die <code>pattern</code> entsprechen. Der Defaultwert von <code>pattern</code> akzeptiert alles.	-	V1.0
3	core	IntegerValidator	-	Ausprägung des RegexValidators. <code>pattern</code> wurde so gesetzt das jeder Integer akzeptiert wird. Es existieren keine Maximal- und Minimalwerte.	-	V1.0
4	core	DecimalValidator	-	Ausprägung des RegexValidators. <code>pattern</code> wurde so gesetzt das jede Dezimalzahl akzeptiert wird. Es existieren keine Maximal- und Minimalwerte.	-	V1.0
5	core	ScriptValidator	<code>script</code> - Ein Stück JavaScript-Code	Der ScriptValidator verfügt über ein Stück JavaScript-Code welches ausgeführt wird um Werte auf Gültigkeit zu überprüfen. Die letzte Anweisung im JavaScript-Code muss einen boolschen Wert zurückgeben. Defaultmäßig ist kein Script gesetzt.	-	V1.0
6	core	MonthValidator	-	Ausprägung des ScriptValidators. <code>script</code> wurde so gesetzt, dass nur Integerwerte im Wertebereich [1, 12] akzeptiert werden.	-	V1.0
7	core	DayValidator	-	Ausprägung des ScriptValidators. <code>script</code> wurde so gesetzt, dass nur Integerwerte im Wertebereich [1, 31] akzeptiert werden.	-	V1.0
8	core	HourValidator	-	Ausprägung des ScriptValidators. <code>script</code> wurde so gesetzt, dass nur Integerwerte im Wertebereich [0, 23] akzeptiert werden.	-	V1.0

9	core	Minutes Validator	-	Ausprägung des ScriptValidators. <code>script</code> wurde so gesetzt, dass nur Integerwerte im Wertebereich [0, 59] akzeptiert werden.	-	V1.0
10	core	Seconds Validator	-	Ausprägung des ScriptValidators. <code>script</code> wurde so gesetzt, dass nur Integerwerte im Wertebereich [0, 59] akzeptiert werden.	-	V1.0
11	core	DateValidator	-	Ausprägung des ScriptValidators. <code>script</code> wurde so gesetzt, dass für eine Variable vom Typ Date (s.u.) geprüft wird, ob das Datum gültig ist (bpsw. Prüfung für Schaltjahre).	-	V1.0
12	core	Unsigned Integer Validator	-	Ausprägung des ScriptValidators. <code>script</code> wurde so gesetzt, dass nur positive Integerwerte akzeptiert werden.	-	V2.0

## Datentypen

	Namespace	Name	Supertyp (en)	Attribut (e)	Defaultwert	Validator (en)	Beschreibung	Änderungen in Version (en)	Hinzugefügt in Version
1	core	Text	-	-	<i>Leerer Text</i>	-	Elementarer primitiver Datentyp der von vielen anderen Datentypen als Supertyp verwendet wird	-	V1.0
2	core	Composite	-	-	-	VoidValidator	Leerer zusammengesetzter Datentyp. Wird von vielen weiteren zusammengesetzten Datentypen als Supertyp verwendet	-	V1.0
3	core	Integer	Text	-	0	IntegerValidator	Datentyp für positive und negative Ganzzahlen.	-	V1.0
4	core	Decimal	Text	-	0.0	DecimalValidator	Datentyp für positive und negative Dezimalzahlen.	-	V1.0
5	core	Month	Integer	-	1	MonthValidator	Darstellung des Monats als positiven, numerischen Ganzzahlwert.	-	V1.0
6	core	Day	Integer	-	1	DayValidator	Darstellung des Tags (im Monat, nicht Wochentag) als positiven, numerischen Ganzzahlwert.	-	V1.0
7	core	Hour	Integer	-	0	HourValidator	Darstellung der Stunde des Tages als positiven, numerischen Ganzzahlwert.	-	V1.0
8	core	Minute	Integer	-	0	MinuteValidator	Darstellung der Minuten in einer Stunde als positiven, numerischen Ganzzahlwert.	-	V1.0
9	core	Second	Integer	-	0	SecondValidator	Darstellung der Sekunden in einer Minute als positiven, numerischen Ganzzahlwert.	-	V1.0
10	core	Date	Composite	day - Day month - Month year - Integer - Defaultwert 1980	01. 01. 1980	DateValidator	Numerische Darstellung eines validen Datums.	-	V1.0
11	core	Time	Composite	hour - Hour minute - Minute second - Second	00: 00: 00	-	Numerische Darstellung einer Uhrzeit.	-	V1.0
12	core	DateTime	Composite	date - Date time - Time	00:00:00 01.01.1980	-	Darstellung eines Datums als Kombination aus Datum und Uhrzeit.	-	V1.0
13	<del>core</del>	<del>Address</del>	<del>Composite</del>	<del>street - Text city - Text</del>	<del>Leerer Text Leerer Text</del>	-	<del>Eine Adresse, bestehend aus Angaben zur Straße und Stadt.</del>  Dieser Datentyp war nur für Tests in der Version 1.0 enthalten und ist in seiner bestehenden Form nicht mehr Bestandteil des <code>core</code> -Moduls.	V2.0	V1.0
14	<del>core</del>	<del>Person</del>	<del>Composite</del>	<del>forename - Text surname - Text birthday - Date address - Address</del>	<del>Leerer Text Leerer Text 01.01.1980 Leerer Text</del>	-	<del>Repräsentation einer Person mit Vor- und Nachnamen, einem Geburtstag und einer Adresse.</del>  Dieser Datentyp war nur für Tests in der Version 1.0 enthalten und ist in seiner bestehenden Form nicht mehr Bestandteil des <code>core</code> -Moduls.	V.20	V1.0

15	core	Student	Person	studentNumber - Text course - Text matriculationDate - Date	Leerer Text Leerer Text 01.01.1980	-	Student als Erweiterung einer Person. Verfügt über eine Matrikelnummer, einen belegten Kurs und ein Immatrikulationsdatum.	-	V1.0
16	core	Error	Text	-	An error occurred.	-	Wenn bei der Ausführung einer Data App Fehler passieren (bspw. die Validierung von Policies schlägt fehl), wird die entsprechende Fehlermeldung in diesem Datentyp verpackt und an den Benutzer zurückgegeben.	-	V2.0
17	core	Tag	Text	-	Leerer Text	-	Eine Data App kann über eine beliebige Anzahl von Tags verfügen, welche sie näher beschreibt (bspw. 'SYSTEM_ADAPTER').	-	V2.0
18	core	Username	Text	-	Leerer Text	-	Jeder Aufruf einer Data App wird von einem bestimmten User durchgeführt, welcher in diesem Datentyp gespeichert wird.	-	V2.0
19	core	Userrole	Text	-	Leerer Text	-	Jeder Nutzer hat eine Menge von Rollen, welche in diesem Datentyp gespeichert werden.	-	V2.0
20	core	URL	Text	-	Leerer Text	-	Die URLs (Uniform Resource Locator) sind eine Teilmenge der URIs und werden zur Identifikation von Ressourcen im Internet verwendet.	-	V2.0
21	core	HttpMethod	Text	-	Leerer Text	-	Wird eine URL aufgerufen, muss die Methode spezifiziert werden (GET, POST, ...).	-	V2.0
22	core	HttpContentType	Text	-	Leerer Text	-	MIME-Type um den Payload eines URL-Aufrufs zu klassifizieren.	-	V2.0
23	core	HttpAcceptedTypes	HttpContentType	-	Leerer Text	-	Komma-getrennte MIME-Type Liste, welche akzeptierte Rückgabewerte eines URL-Aufrufs angibt.	-	V2.0
24	core	Hostname	Text	-	Leerer Text	-	Eindeutige/r Adresse/Name eines Servers. Der Server muss unter diesem Namen erreichbar sein.	-	V2.0
25	core	ServerPort	Text	-	Leerer Text	-	Port unter dem ein bestimmter Dienst eines Servers erreichbar ist.	-	V2.0
26	IDS	Infomodel	Text	-	Leerer Text	-	JSON-Representation des IDS-Infomodels.	-	V2.0
27	IDS	Geolocation	URL	-	Leerer Text	-	GeoURL, welche im Rahmen des IDS-Infomodels zur Positionsbestimmung verwendet wird.	-	V2.0

## Aktivitäten

	Namespace	Name	Eingabeparameter	Ausgabeparameter	Beschreibung	Änderungen in Version (en)	Hinzugefügt in Version
1	core	PrintToConsoleActivity	text - Text	-	Schreibt den Inhalt von text in die Standardausgabe der ausführenden Data App.	-	V1.0
2	core	WriteFileActivity	filePath - Text content - Text	-	Schreibt den kompletten Inhalt von content in die Datei filePath.	-	V1.0
3	core	ReadFileActivity	filePath - Text	content - Text	Liest den gesamten Inhalt der Datei filePath und schreibt ihn in content.	-	V1.0
4	core	ExecCmd	cmd - Text args - Text	returnValue - Text	Erlaubt es eine Kommandozeile (Applikation + Argumente) an das ausführende Betriebssystem abzusetzen.  Die Ausführungsergebnisse stehen innerhalb der Data App zur Verfügung.	-	V2.0
5	core	PerformRest	proxyHost - Hostname proxyPort - ServerPort url - URL httpMethod - HttpMethod contentType - HttpContentType acceptedTypes - HttpAcceptedTypes body - Text	response - Text	Erlaubt es eine gegebene URL mit definierter HttpMethod, Content-Type und Accepted Attributen aufzurufen. Der Payload der Nachricht kann frei gewählt werden. Es ist möglich einen Proxy zu konfigurieren, sollte dies notwendig sein.  Die Rückgabergebnisse des Aufrufs stehen innerhalb der Data App zur Verfügung.	-	V2.0

## Policies

	Namespace	Name	Typ	Eingabeparameter	Abhängigkeiten	Beschreibung	Änderungen in Version (en)	Hinzugefügt in Version
1	core	UseNotBeforeTimestamp	Constraint	timestamp - DateTime	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur nach einem gegebenem Zeitpunkt	-	V1.0
2	core	UseNotAfterTimestamp	Constraint	timestamp - DateTime	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur vor einem gegebenem Zeitpunkt	-	V1.0
3	core	AllowedTimeInterval	Policy	-	UseNotBeforeTimestamp UseNotAfterTimestamp	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur in einem definierten Zeitintervall	-	V1.0
4	core	MaxLength	Constraint	MaxLength - UnsignedInteger content - Text	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur , wenn die Länge eines bestimmten Textes einen gegebenen Maximalwert nicht überschreitet.	-	V2.0
5	core	RequireTag	Constraint	tag - Tag	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn die Data App über ein bestimmtes Tag verfügt.	-	V2.0
6	core	DenyTag	Constraint	tag - Tag	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn die Data App nicht über ein bestimmtes Tag verfügt.	-	V2.0
7	core	RequireUsername	Constraint	username - Username	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn ein bestimmter User den Zugriff initiiert hat. Die Quelle für den Nutzernamen (bspw. OS oder JWT) hängt von der Art der Data App ab.	-	V2.0
8	core	DenyUsername	Constraint	username - Username	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn ein bestimmter User den Zugriff nicht initiiert hat. Die Quelle für den Nutzernamen (bspw. OS oder JWT) hängt von der Art der Data App ab.	-	V2.0
9	core	RequireUsernameOs	Constraint	username - Username	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn ein bestimmter User den Zugriff initiiert hat. Der Benutzername wird aus den OS Informationen geholt.	-	V2.0
10	core	DenyUsernameOs	Constraint	username - Username	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn ein bestimmter User den Zugriff nicht initiiert hat. Der Benutzername wird aus den OS Informationen geholt.	-	V2.0
11	core	RequireUsernameJwt	Constraint	username - Username	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn ein bestimmter User den Zugriff initiiert hat. Der Benutzername wird aus dem JWT, welches dem Request an die Data App angehängt ist, geholt.	-	V2.0
12	core	DenyUsernameJwt	Constraint	username - Username	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn ein bestimmter User den Zugriff nicht initiiert hat. Der Benutzername wird aus dem JWT, welches dem Request an die Data App angehängt ist, geholt.	-	V2.0
13	core	RequireRole	Constraint	role - Userrole	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn der aktuelle Benutzer eine bestimmte Rolle hat. Die Quelle für die Rolleninformationen (bspw. OS oder JWT) hängt von der Art der Data App ab.	-	V2.0
14	core	DenyRole	Constraint	role - Userrole	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn der aktuelle Benutzer eine bestimmte Rolle nicht hat. Die Quelle für die Rolleninformationen (bspw. OS oder JWT) hängt von der Art der Data App ab.	-	V2.0
15	core	RequireRoleOs	Constraint	role - Userrole	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn der aktuelle Benutzer eine bestimmte Rolle hat. Die aktuellen Benutzerrollen werden aus den OS Informationen geholt.	-	V2.0
16	core	DenyRoleOs	Constraint	role - Userrole	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn der aktuelle Benutzer eine bestimmte Rolle nicht hat. Die aktuellen Benutzerrollen werden aus den OS Informationen geholt.	-	V2.0
17	core	RequireRoleJwt	Constraint	role - Userrole	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn der aktuelle Benutzer eine bestimmte Rolle hat. Die aktuellen Benutzerrollen werden aus dem JWT, welches dem Request an die Data App angehängt ist, geholt.	-	V2.0
18	core	DenyRoleJwt	Constraint	role - Userrole	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur, wenn der aktuelle Benutzer eine bestimmte Rolle nicht hat. Die aktuellen Benutzerrollen werden aus dem JWT, welches dem Request an die Data App angehängt ist, geholt.	-	V2.0
19	IDS	RequireGeolocation	Constraint	infomodel - IDS. Infomodel  location - IDS. Geolocation	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur	-	V2.0

20	IDS	DenyGeolocation	Constraint	infomodel - IDS. Infomodel  location - IDS. Geolocation	-	Erlaubt die Ausführung einer Aktivität/Zugriff auf Daten nur	-	V2.0
----	-----	-----------------	------------	---	---	--	---	------

## Kontextmodule

### Mappings

Abbildung von	Abbildung auf	Bedingung
UserInformation	OsUserInformation	Dieses Mapping wird standardmäßig angewandt, sofern kein anderes Mapping für ein Modul, welches Benutzerinformationen bereitstellt, vorhanden ist.
UserInformation	JWTUserInformation	Dieses Mapping wird immer dann angewandt, wenn es sich bei der aktuellen Data App um eine REST-Data App handelt.

### Module

	Name	enthaltene Elemente	Beschreibung	Änderungen in Version (en)	Hinzugefügt in Version
1	TagsKontextModule	-	Das Modul enthält keine Elemente, die immer vorhanden sind. Stattdessen werden alle Tags über die eine Data App verfügt in diesem Modul als ReadOnlyEntity abgelegt.	-	V2.0
2	JWTUserInformation	username - ReadWriteEntity  userroles - ReadWriteEntity	Das Modul enthält Informationen über den Nutzer, der seinen JWT als Teil des aktuellen Request an die Data App übermittelt. Die Daten werden bei jedem Request aktualisiert.	-	V2.0
3	OsUserInformation	username - ReadOnlyEntity  userroles - ReadOnlyEntity	Das Modul enthält Informationen über den aktuellen Betriebssystemnutzer, welcher die Data App ausführt. Die Daten werden beim ersten Start vom System abgefragt und sind danach nicht mehr änderbar.	-	V2.0